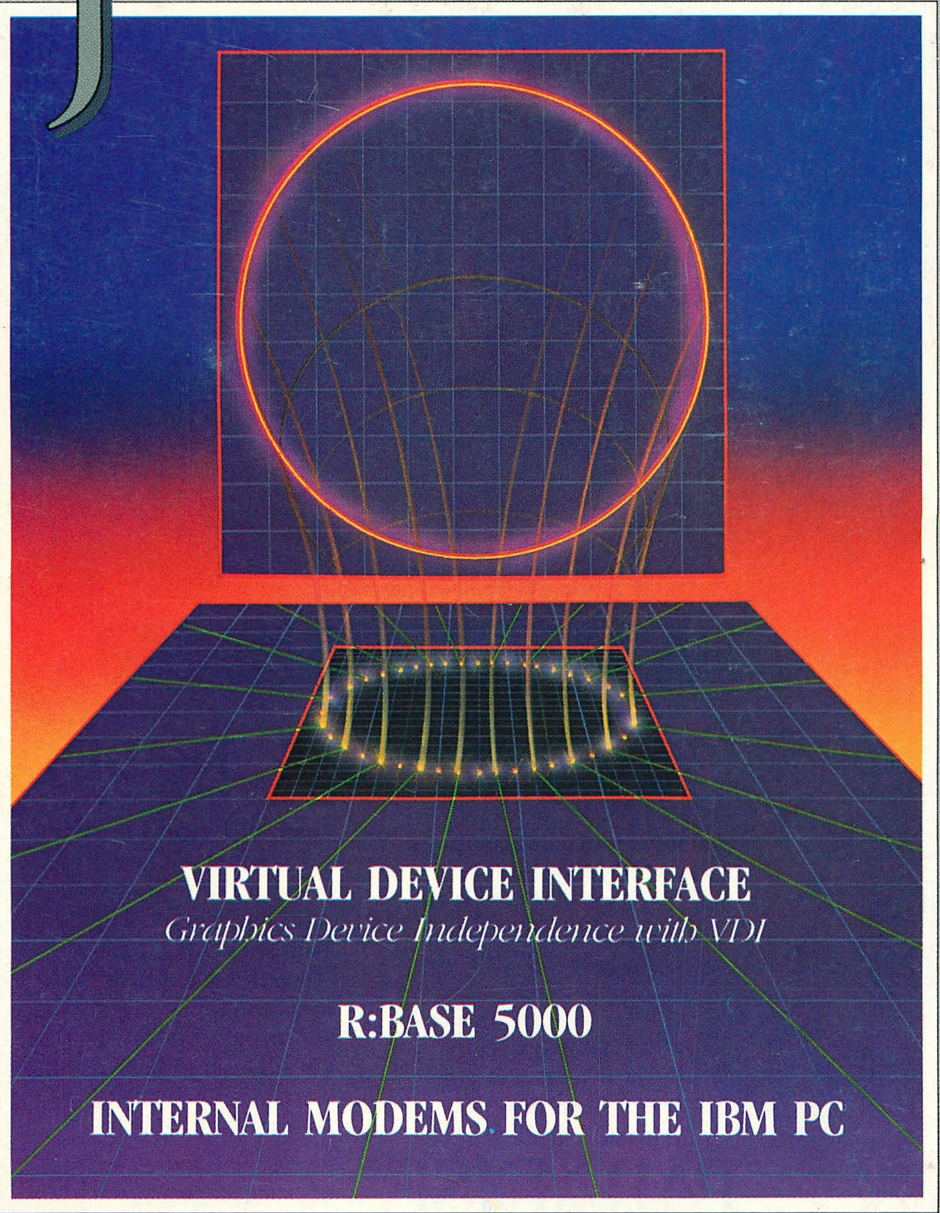
SEPTEMBER 1985 VOL. 3, No. 9 $3.95

FOR IBM PERSONAL COMPUTER USERS

PC TECH JOURNAL



VIRTUAL DEVICE INTERFACE

Graphics Device Independence with VDI

R:BASE 5000

INTERNAL MODEMS FOR THE IBM PC

"Nature Imposes Few Restrictions on Those Daring Enough to Lead"

ZIM is a fully integrated fourth generation application development system designed for leading system integrators and corporate and independent applications developers.

*COMPLETE DEVELOPMENT ENVIRONMENT
- -Report Writer
- -Forms Painter and Manager
- -Data Dictionary
- -Application Generator
- -Non-procedural Programming Language
- -Compiler
- -C Language Interface
- -Runtime System

*POST RELATIONAL
- -Entity Relationship Model
- -Powerful extension of Relational Model

*MAINFRAME POWER, FUNCTIONALITY AND PERFORMANCE

*APPLICATIONS PORTABILITY
- -MS-DOS, UNIX, XENIX, and QNX

*MULTI-USER
- -Full transaction processing control

*NETWORKING

*APPLICATIONS LIMITED ONLY BY HARDWARE

*BUILT-IN STRATEGY OPTIMIZER

*ENGLISH-LIKE LANGUAGE

*QUALITY PRODUCT SUPPORT

ZIM is a mainframe system that runs on micro-computers and on super micro-computers. If you want mainframe power, speed, flexibility and freedom from arbitrary limitations all at a micro price, talk to us about an evaluation system.
Dealer inquiries are welcome.

ZIM
The Information Interface

ZANTHE

1785 Woodward Dr., Ottawa, Ontario
K2C 0R1 (613)727-1397

MS-DOS and XENIX are Microsoft Corp. trademarks. UNIX is an AT&T trademark. QNX is a Quantum Software Systems trademark.

CIRCLE NO. 148 ON READER SERVICE CARD

If you're tired of waiting, you're using the wrong file manager.

Be fast. Btrieve.™

If batch jobs and reports turn waiting time to nighttime, then wake up! You're using the wrong file manager.

Btrieve™ file management hates waiting as much as you do. It's written in assembly language especially for the IBM®PC. And based on b-tree file indexing, with automatic balancing for access speed that won't degrade as your database grows. With Btrieve, your applications always run fast. So you'll be out the door faster.

The standard for networking. When your application requires multi-user file sharing, Btrieve/N (network version) sets the standard for the industry's most popular LANs: IBM's PC Network, Netware™, Davong MultiLink™, Omninet™, PC Net™, Ether Series™, Nestar™, and NetOne™. Btrieve/N offers *safe* network file management that coordinates simultaneous updates and prevents lost data.

Automatic file recovery. Btrieve provides automatic file recovery after a system crash, so accidents and power failures don't turn into database disasters. Your Btrieve data always comes back intact.

Fully-relational data management. Using SoftCraft's entire family of products gives you a complete, fully-relational database management system. Rtrieve™ adds report writing capabilities for generating the reports you need. Xtrieve™ speeds users through database queries with interactive, on-screen menus—no command language or special syntax.

For professional programmers. Btrieve is the fast, reliable answer for all your application development. In any development language—BASIC, Pascal, COBOL, C, FORTRAN, and APL. With multikey access to records. Unlimited records per file. Duplicate, modifiable, and segmented keys. With Btrieve, you can develop better applications faster. And know they'll run—*fast.*

Suggested retail prices: Btrieve, $245; Btrieve/N, $595; Xtrieve, $195; Xtrieve/N, $395; Rtrieve, $85; Rtrieve/N, $175. Requires PC-DOS or MS™-DOS 1.X, 2.X, or 3.X.

Btrieve, Xtrieve, and Rtrieve; IBM; Netware: Davong MultiLink; Omninet; PC Net; EtherSeries; Nestar; NetOne; and MS are trademarks of SoftCraft Inc.; International Business Machines; Novell Data Systems; Davong Systems Inc.; Corvus Systems; Orchid Technology; 3Com Corp.; Nestar Systems Inc.; Ungermann-Bass; and Microsoft Inc.

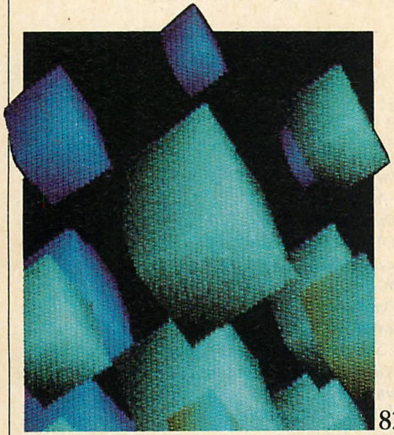
SoftCraft Inc.

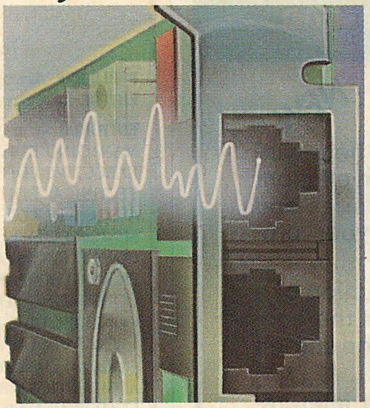
P. O. Box 9802 #917
Austin, Texas 78766
(512) 346-8380 Telex 358 200

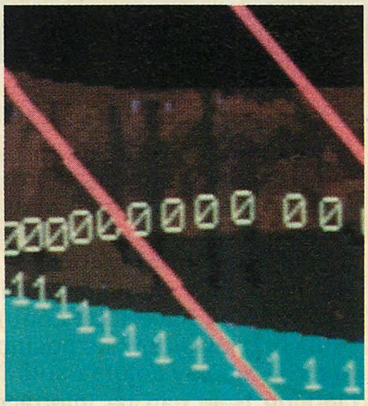
CIRCLE NO. 201 ON READER SERVICE CARD

PC TECH JOURNAL

SEPTEMBER 1985 VOLUME 3, NUMBER 9


82


60


104

Cover illustration · Robert Pasternack

PC TECH JOURNAL

VOL. 3, NO. 9

PUBLISHER: *Newton Barrett*
EDITOR: *Will Fastie*

EDITORIAL
MANAGING EDITOR: *Marjory Spraycar*
EXECUTIVE EDITOR: *Julie Anderson*
TECHNICAL EDITOR: *Jeff Duntemann*
ASSOCIATE TECHNICAL EDITOR: *Dan Beale*
SENIOR COPY EDITOR: *Susan Holly*
COPY EDITOR: *Gail Shaffer*
PROOFREADER: *Kathleen Peddicord*
CONSULTING EDITOR: *Thomas V. Hoffmann*
CONTRIBUTING EDITORS: *Steven Armbrust, Don Awalt, Richard Foard, Ted Forgeron, Arthur A. Gleckler, Augie Hansen, Ted Mirecki, William H. Murray, Max Stul Oppenheimer*
ADMINISTRATIVE ASSISTANT: *Paula Lamberti*
EDITORIAL ASSISTANT: *Carole Autenzio*

ART & PRODUCTION
ART DIRECTOR: *Ina Saltz*
ASSOCIATE ART DIRECTOR: *Sharon Reuter*
ASSISTANT ART DIRECTOR: *Sandra Ray*
ART SECRETARY: *Sarah Jones*
ADVERTISING PRODUCTION MANAGER: *Patricia White*
EDITORIAL PRODUCTION COORDINATOR: *David K. Chipps*

ADVERTISING SALES
MARKETING COORDINATOR: *Julie Henderson*
ADVERTISING COORDINATOR: *Michele Fischetti*
DISTRICT MANAGERS: *Rita Burke, Jan Schultz—East Coast; Lisa Kampfmann—Midwest; Ted Bahr, Bill Bush, Phyllis Egan—West Coast*
ACCOUNT REPRESENTATIVES: *Pat Toohey—East Coast; Rosemarie Caruso—Midwest and East Coast; Pam Sigal, Arlene Steadman—West Coast*

CIRCULATION
CIRCULATION MANAGER: *Shane Boel*
CIRCULATION SALES DEVELOPMENT: *Daniel Rosensweig*
MEDIA MANAGER: *Melinda Kendall*

COMPUTER PUBLICATIONS DIVISION
PRESIDENT: *Kenneth H. Koppel*
VICE PRESIDENT, Editorial: *Jonathan D. Lazarus*
VICE PRESIDENT, Production: *Baird Davis*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Circulation: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT: *Hugh Tietjen*
MARKETING MANAGER: *Ronnie Sonnenberg*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
BUSINESS MANAGER: *Gary A. Gustafson*
EDITORIAL DIRECTOR: *Ernest F. Baxter*

ZIFF-DAVIS PUBLISHING
PRESIDENT, *Richard P. Friese;* SENIOR VICE PRESIDENT, *Kenneth H. Koppel;* SENIOR VICE PRESIDENT, *Philip Sine;* VICE PRESIDENT, *William L. Phillips;* TREASURER, *Selwyn Taubman;* SECRETARY, *Bertram A. Abrams*

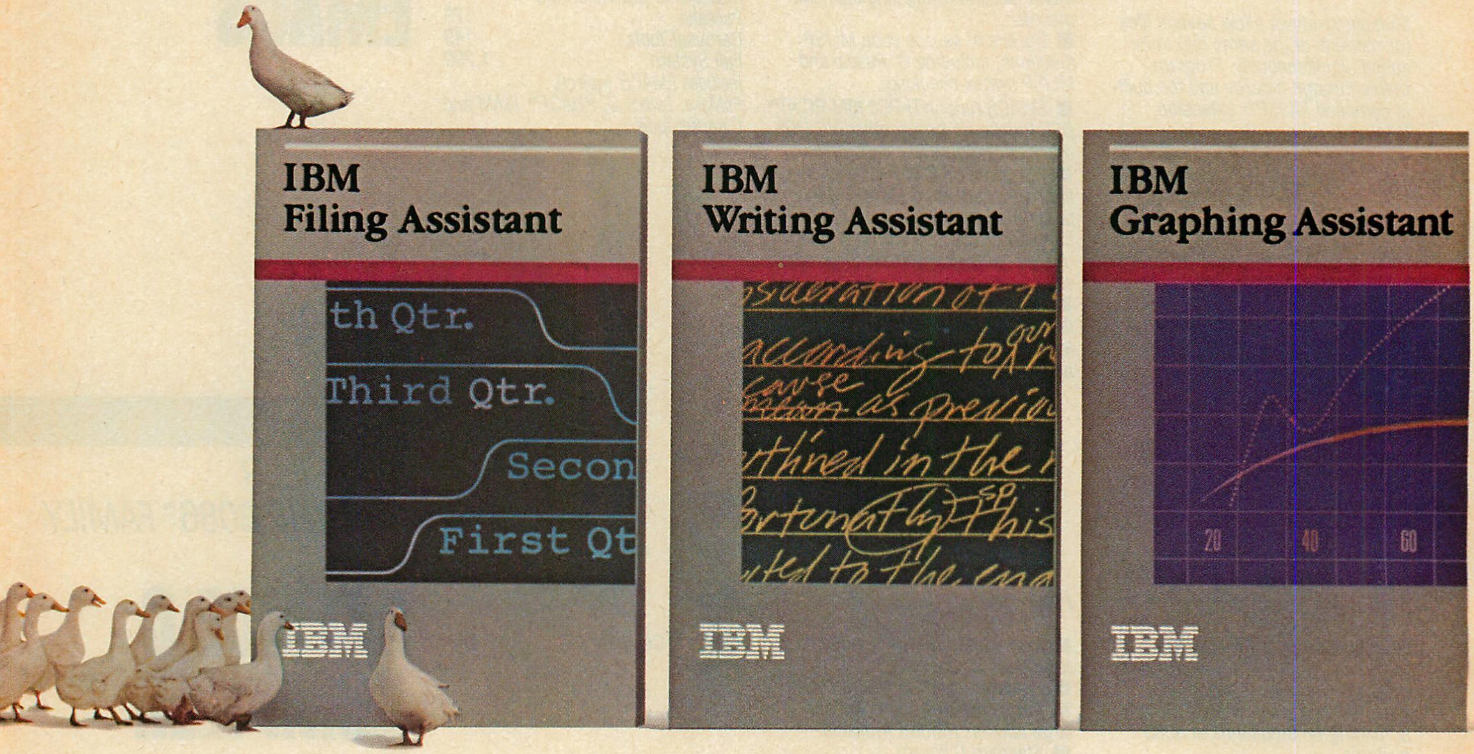
EDITORIAL OFFICE
PC TECH JOURNAL, The World Trade Center, Suite 211, Baltimore, MD 21202. 301/576-0770. MCIMail: PCTECH. PCTECHline: 301/576-PCTJ.

ADVERTISING OFFICES
(East Coast/Midwest) One Park Ave., New York, NY 10016. 212/503-5185. (West Coast) 3460 Wilshire Blvd., Los Angeles, CA 90010. 213/387-2100; 11 Davis Drive, Belmont, CA 94002. 415/598-2290.

SUBSCRIPTION INQUIRIES
PC TECH JOURNAL, P.O. Box 2968, Boulder, CO 80321. Subscription service: 303/447-9330. Back issues: send $7.00/copy to *PC TECH JOURNAL*, Box CN, 1914, Morristown, NJ 07960.

PC TECH JOURNAL (ISSN 0738-0194) is published monthly, $29.97 for one year, $52.97 for two years, $69.97 for three years. Additional postage $12 for Canada & Foreign by Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. Second-Class Postage paid at New York, NY and at additional mailing offices. POSTMASTER: Send address changes or subscription inquiries to P.O. Box 2968, Boulder, CO 80321.

BPA SELECTED MARKET AUDIT DIVISION



How to get all your

Now you can easily control even hard-to-manage facts and figures with the Assistant Series from IBM.

If you line up the right software, getting your ducks in a row is not only possible, but easy.

There's a program in IBM's Assistant Series to handle each of the most important jobs people are using personal computers for these days: spreadsheets, word processing, and database management.

But unlike software programs that can't share information with other programs, the IBM Assistants work together as a team, as well as alone.

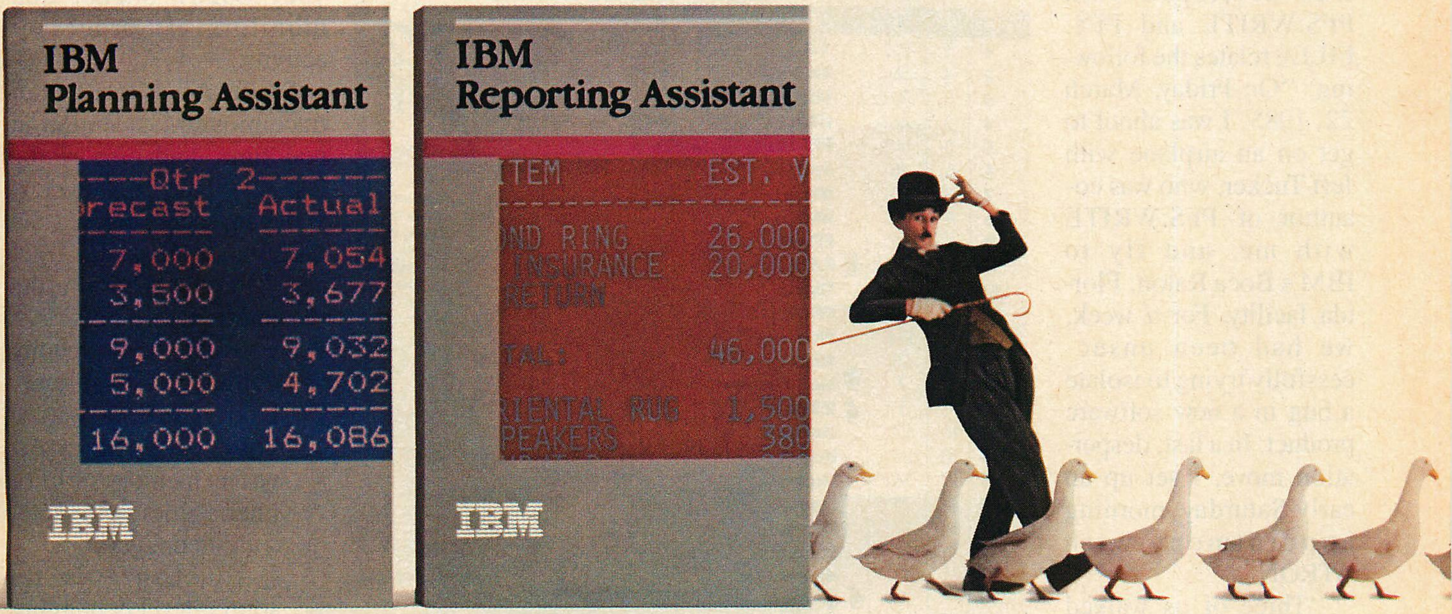
And unlike other programs that can be challenging to learn, all the complicated inner workings of these programs have been hidden from view. So using them is almost intuitive.

Filing Assistant is a database program that allows you to organize and store information in any format that suits you, and retrieve it instantaneously, according to your own specifications.

Writing Assistant is a word processing program that shows you on the screen just what your words will look like on paper. It helps correct misspelled words and even lets you insert forecasts or graphs from other Assistants into your text.

Graphing Assistant turns numbers into pictures. Like all the Assistants, it accepts information from the keyboard,

*IBM Product Center prices.



ducks in a row.

or directly from Filing Assistant. So you can see what the bottom line looks like as a pie chart, a bar chart or a line chart. Or all three. In minutes.

Planning Assistant is a powerful new spreadsheet that replaces the complex commands used in earlier programs with plain English (TOTAL instead of C21+D21+E21+F21). It's smart enough to widen columns automatically to fit your entries. It will even fill in the blank after "Dec 85" with "Jan 86."

Reporting Assistant helps you create customized reports (complete with totals, subtotals and averages) of your Filing Assistant data.

If you generate so many Writing Assistant files that you can't remember them all by name, there's *Document Retrieval Assistant* to help you find them anyway.

And there's *Mainframe Communication Assistant* to do just what its name suggests: link your PC to your company's mainframe computer.

To assist your pocketbook, most IBM Assistants cost less than $150.*

Any Authorized IBM Personal Computer Dealer or IBM Product Center can give you a demonstration of the Assistant Series. Just seeing them in action may make you feel more organized. For the location of the store nearest you and a free brochure, call 1-800-447-4700. In Alaska or Hawaii, 1-800-447-8090.

IBM®

Personal Computer Software

Little Tramp character licensed by Bubbles Inc., s.a.

CIRCLE NO. 150 ON READER SERVICE CARD

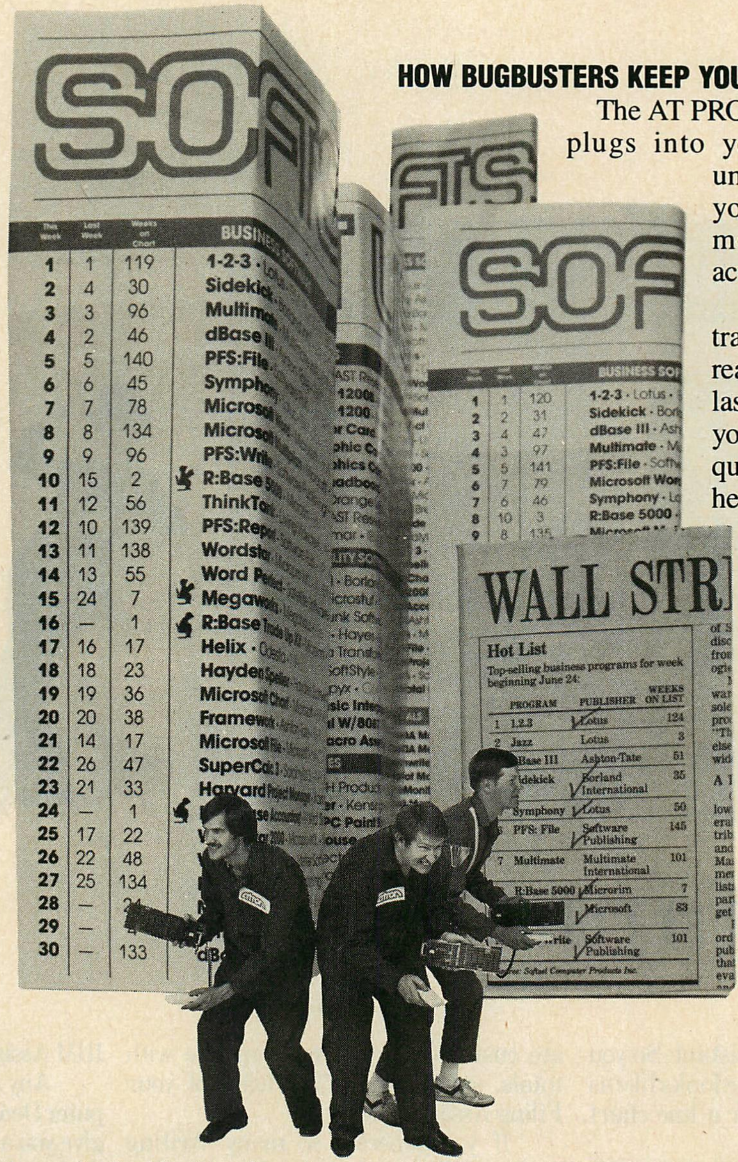
ATRON'S PC/AT BUGBUSTERS

A BUGBUSTER STORY

Brad Crain, a project manager at Software Publishing (the people who developed both PFS:WRITE and PFS:FILE), relates the following: "On Friday, March 22, 1985, I was about to get on an airplane with Jeff Tucker, who was co-author of PFS:WRITE with me, and fly to IBM's Boca Raton, Florida facility. For a week, we had been unsuccessfully trying to isolate a bug in a new software product. In a last, desperation move, I set up an early-Saturday morning appointment with ATRON.

"Three of us walked through ATRON's door at 8:00 the next morning. Using ATRON's hardware-assisted debugging tools, we had the problem identified and fixed by 10:30AM."

Mr. Crain concludes: "We'd never have found the bug with mere software debuggers, which have the bad habit of getting over-written by the very bugs they're trying to find. It doesn't surprise me that almost all the top-selling software packages were written by ATRON customers. Now that they've broadened their PC family of debuggers to include a PC/AT debugging tool, those of us seriously into 80286 development are greatly relieved."



ARE YOU TRYING TO DO SOMETHING SCAREY?

Like developing your AT-based software product in the dark? Without professional debugging tools?

Seven of the ten top-selling software packages listed by the *THE WALL STREET JOURNAL** were produced by ATRON customers. The PC PROBE™ bugbuster ($1595) accounts for much of this success. Now that the PC/AT is the new standard for advanced commercial and scientific development, ATRON is proud to announce the AT PROBE™ bugbuster ($2495). It has even more debugging capabilities than the PC Probe.

HOW BUGBUSTERS KEEP YOU FROM GETTING SLIMED

The AT PROBE is a circuit board that plugs into your PC/AT. It has an umbilical which plugs into your 80287 socket and monitors all processor activity.

Since AT PROBE can trace program execution in real time, and display the last 2048 memory cycles, you can easily answer the questions: "How did I get here?" and "What *are* the interrupts doing?"

It can solve spooky debugging problems. Like finding where your program over-writes memory or I/O - impossible with software debuggers.

You can even do source-level debugging in your favorite language, like C, Pascal or assembler. And after your application is debugged, the AT PROBE's performance-measurement software can isolate your application's bottlenecks.

Finally, the AT PROBE has its own 1-MByte of memory. Hidden and write-protected. How else could you develop that really large program, where the symbol table would otherwise demand most of your PC/AT memory.

BORLAND'S PHILIPPE KAHN: "THERE WOULDN'T BE A SIDEKICK™ WITHOUT ATRON'S DEBUGGERS."

So why waste more time reading though your program listing for the ten thousandth time, trying to find why your program starts howling with every full moon. Be like BORLAND, get your Atron bugbuster today and bust bugs tomorrow.

atron
THE DEBUGGER COMPANY
20665 Fourth Street • Saratoga, CA 95070 408/741-5900

*WSJ, June 24, 1985, reporting Softsel figures. © 1985 by ATRON. PC PROBE™ and AT PROBE™ ATRON. SIDEKICK™ Borland. IBM Corp. owns numerous trademarks. Ad by TRBA.

CIRCLE NO. 203 ON READER SERVICE CARD

User's Rights

And the evils of copy protection

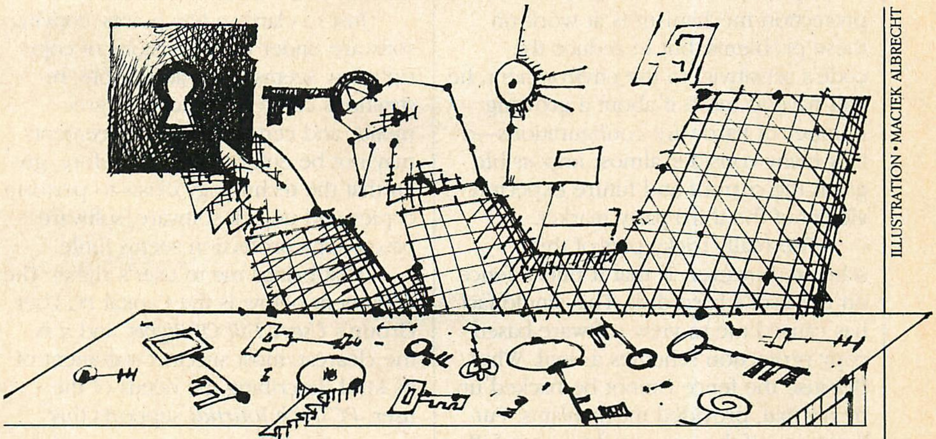


ILLUSTRATION • MACIEK ALBRECHT

I first used this space to comment about software protection in the March 1984 issue. Since then, not much has happened to improve the situation. The hardware protection scheme proposed in that editorial has not emerged, although ADPASO has circulated a draft proposal for a hardware key. The software protection schemes have become more complex, and the newest one even provides an apparent convenience to the user. Sales of industrial-strength copy programs continue, and new programs are available that claim to perform automatic *ProLok-* or *SuperLockectomies* (a brilliant coinage from former *PC Tech Journal* technical editor Susan Glinert). An unscientific survey of our readership indicates a high degree of familiarity with manual versions of these surgical techniques.

Two views regarding software protection have been traditionally considered as diametrically opposed:

- Users want and demand the ability to run software conveniently and the ability to safeguard their investments, which are sometimes considerable.
- Vendors want and demand the right to a fair price for each copy of their software and the unimpaired ability to prospect and mine the product's potential market to the fullest extent.

Yet, even abusers (vendors and users alike) agree that these are reasonable goals. What becomes the major point of contention is the *method* by which the software vendor decides to protect the product's market; it is here that a vicious battle line is drawn.

Resolving this dilemma in my own mind has taken a long time, but I have concluded that the current methods of software protection, commonly known as copy protection, are not in anyone's best interest. This is based on the fact that the user, who foots the bill for a product and its vendor's ability to survive, pays an additional, significant cost in terms of inconvenience and risk of software loss; yet it is the user who is supposed to gain efficiency and productivity by using the software.

Software-effected copy protection has been directed completely by the technology of the desktop computer. The first schemes simply "nibbled" the diskette in such a way that the standard utility programs provided with the operating systems were unable to read certain parts, or *fences*, on the diskette. Information was encoded into the fence that only the applications program knew how to read; when the program began execution, it checked for the presence of the proper fence. The user was forced to install the vendor's master diskette to run a program, but this was not an inconvenience because, at the time, that was the *only* way to run a program anyway. The advent of hard disks made this technique extremely annoying and created the first major hubbub of the copy-protection debacle.

The problem was simple. With the hard disk, the user made a quantum leap in the convenience of computer use. Programs had to be slightly revised to allow the software to be copied to the hard disk (or another floppy) but required the presence of the floppy in order to get the program running. The concept of the *key disk* was born. Advantage: programs loaded faster, overlaid programs ran faster, and diskette swapping (for programs with lots of files) was eliminated. Problem: the user still had to go through finding and inserting the key disk. Result: no gain in time, plenty of irritation.

The industry heard this squawk from the users, and a new form of copy protection was developed. In this scheme, the application is installed with a special program that keeps track (on the master system diskette) of how many times the software has been installed. The program places a fence on the hard disk and hides its execution control information there, making it completely independent of the master system diskette and able to be run without a key disk. The program can be moved from one computer to another by running an un-install procedure, effectively copying the program back onto the master system diskette and adjusting the hidden records. Friendlier still, the technique allows the vendor to supply more than one "install," providing backup in case of catastrophe or the opportunity to install the software on more than one computer. As a final touch, the technique allows multiple applications to record their control information in the same fence.

The idea sounds good, and it worked very smoothly at first, but on

closer examination it is insidious and problematic. First, the protection code has a hair trigger and is extremely sensitive to the environment. A number of programs that installed well on my XT refused to go on the AT. An update to the software cured many of these problems (presumably the code was tuned to reduce its sensitivity), but some packages still had problems. In one case, two or three tries were required to get the program installed. After that, the program ran only after three or four attempts; there were apparent problems reading the fence. The vendor of the protection mechanism is at work on these problems; but to reduce the code's sensitivity to the environment, he will have to teach it about a growing number of hardware configurations—a task many consider almost impossible given the current and future expected vitality of the third-party market.

The really tricky part of this scheme, however, is that it puts a fence on the user's hard disk. This single fact has caused me to view software-based copy protection schemes as evil. Why? Because the fence cannot be backed up by normal, provided mechanisms. Furthermore, if the user *could* successfully get the program off the hard disk, he would have broken protection and thus created the opportunity to reinstall on more than one computer, the very operation the technique is supposed to prevent. Not being able to back up a hard disk is absolutely, totally, completely, and (insert other, all-inclusive adverbs here) unacceptable. True, the programs can be un-installed before the back-up and then put back on the hard disk—nothing like taking a process nobody likes and most don't remember and making it more difficult, more time-consuming.

The bottom line is that all these techniques can be broken and regularly are. Moreover, legal opinion seems to support the user's right to back up and copy as long as copies are not illegally distributed. Legal opinion also says that breaking copy protection to perform such legal copies is defensible (meaning a strong, if not an airtight, case). Better yet, the anti-hacker statutes now on the books in some states and under consideration in others make unauthorized access of computer equipment illegal; the strong argument is made that writing a fence on a hard disk is just such an access. In other words, the user may authorize a software installation but not the writing of data on the disk in such a way that the user is unable reasonably to operate the system thereafter (such as make a back-up).

These legal positions all have to do with United States copyright law and seem to be widely accepted at the moment. A more difficult issue is the software license agreement; it usually contains the language responsible for restricting the user's rights. Shrink-wrapped licenses are discussed in this month's Legal Brief ("Shrink-wrapped Enforcement," p. 177). There is a growing feeling that such licenses are not enforceable; attorney Max Stul Oppenheimer offers some clarification.

Just to clarify a few points: copying software under the provisions of copyright law seems to be acceptable, restrictions come from license agreements, and certain license agreements may not be enforceable. Therefore, given that the technology exists to perform copies and unlock software, software-based copy protection seems futile.

That brings me to user's rights. The document below is the Capital PC User Group's *User's Bill Of Rights*, and it is the clearest, most succinct statement of its kind describing the needs of the user. *PC Tech Journal* supports this document and encourages software vendors to consider it carefully. When doing so, also consider that the vast majority of software users have no desire to deprive any vendor of fair compensation, nor do they desire the demise of the source of support for a product in which they have invested. The time has come for software vendors to develop a closer relationship with end users, to understand more fully their needs and wants. More than timely; it is urgent.

This is a firm stand on the user's side, but it should not be taken as anti-vendor. Quite the contrary: All reasonable users understand the vendor's expectation that users will abide by the law and not violate the vendor's rights. I strongly support the vendor's rights. However, the vendor knows that compliance is virtually impossible to police. Some method is needed to encourage users to comply without unduly hampering their productivity. Software-based systems do not work. The industry must move toward a hardware solution, perhaps similar to the ADAPSO proposal, that effectively keys the software system while providing flexibility, functionality, and convenience.

CAPITAL PC USER GROUP, INC.
BILL OF RIGHTS FOR SOFTWARE USERS

1. Right to Product Quality:
   The user has the right to expect a software product to perform with a level of quality consistent with industry-specified or implied standards.
2. Right of Functionality:
   The user has the right to expect a software product to perform basic functions common to the generic program type, with any exceptions or caveats fully disclosed prior to purchase.
3. Right of Continuous Service:
   The user has the right to have on hand sufficient backup of program packages and authorization devices to continue operations uninterrupted by loss of or damage to the primary package or authorization device.
4. Right to Program Support:
   The user has the right of access to information from the software manufacturer concerning known errors, fixes, or temporary work-arounds. The user also has the right to reasonable updates to the released versions for the correction of errors.
5. Right of System Integration:
   The user has the right to integrate software products into his or her system environment without undue constraints or interference from copy protection, software authorization, or other extraneous program functions.

Approved May 6, 1985, by the Capital PC User Group Board of Directors.

The above version of the Bill of Rights for Software Users was approved by the CPCUG Board of Directors at the May board meeting, and has been distributed to the press, industry, and legislators. This bill is intended to set forth in writing our needs as software users and is not intended to take away the right of software vendors to protect their products or compensation. It is simply time we let the industry leaders know how we need to use their products so they will know better how to interact with us.

Dave Browning
Vendor Relations
Capital PC User Group

NEW!
Retail Price
$895

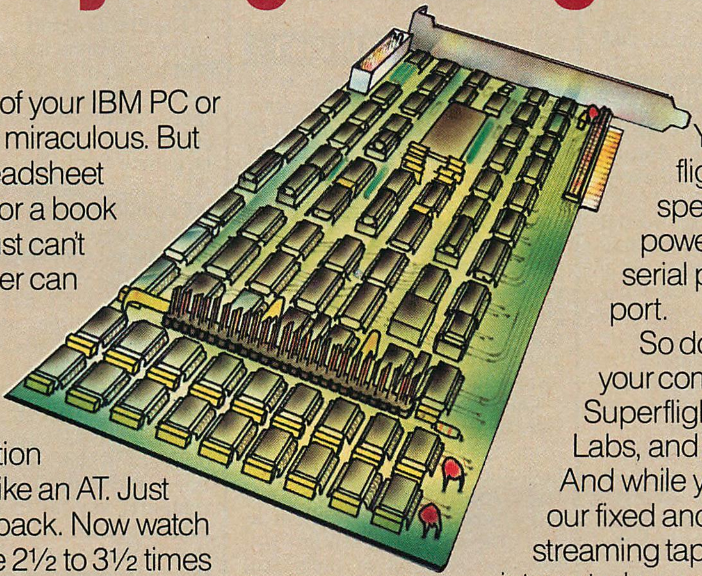
3 simple ways to throw your office into a "wait state."

SORT YOUR BILLS

SEARCH YOUR PAYROLL

SAVE FOR AN IBM AT

1 simple way to get things moving.



At first, the processing speed of your IBM PC or PC/XT seemed nothing short of miraculous. But lately, asking it to do a large spreadsheet seems like asking a first-grader for a book report on War and Peace. You just can't afford to wait that long. But neither can you afford to buy a new PC/AT.

So, what to do?

The answer comes from Kamerman Labs: Superflight.

This remarkable PC acceleration board makes your PC perform like an AT. Just load your software—and stand back. Now watch your programs run an incredible 2½ to 3½ times faster. And let your productivity soar.

Superflight puts a 9.54 MHz Intel 8086 chip and up to 640K of 16-bit high-speed RAM into your PC. Unlike the AT, or other acceleration boards that use an 80186/286 chip, Superflight guarantees compatibility with your software library.

There's still more.

You can expand Superflight to include a high-speed 8087, or a battery-powered clock/calendar, serial port and parallel printer port.

So don't let your PC stunt your company's growth. Get Superflight from Kamerman Labs, and bring it up to speed. And while you're at it, ask about our fixed and removable hard disks, streaming tape backup, and integrated mass storage systems for the IBM PC, XT, AT, and compatibles.

For more information, call toll-free 1-800-522-2237. Ask for Dept. 357. In Oregon and Alaska call 503-626-6877. Ask for Dept. 357. Dealer and OEM inquiries welcome.

Kamerman Labs

8054 S.W. Nimbus Ave., Beaverton, OR 97005
Telex 501 886

Low power

Complete hard disk kit $395

Comes complete with virtually the identical controller that's in the IBM® XT, and Xerox® warranties the hard disk for one year

Guaranteed 100 percent IBM PC compatible

How can we offer this fantastic price? Simple. We buy in such volume that even the most avaricious hard disk businessmen understand they have to give us the best price possible. We could pocket the difference, but we don't.

Instead, we put the extra profit into our testing facilities. That's why Xerox guarantees our $395 10 megabyte hard disk for one year.

Xerox knows, as our customers know, that we have an extensive testing program. Here is what we contribute toward giving you the maximum hard disk performance.

Best Drives Available

First, we buy the best drives available. Sounds trite, doesn't it? I mean, a drive's a drive—right? Hardly. You should see some of the junk we get in our labs. Some have such high failure rates that we even questioned our own $10,000 hard disk tester. But when we tested other manufacturers' drives we were assured that our equipment was fine, which just confirmed that the bad hard disks were not only bad—they were real bad.

But that's just the weeding out process. We then take each drive that we've put through our tester and test it again with the controller you've requested. We call this a "tested pair."

DOS Doesn't Do It

In case you're thinking that all this is an unnecessary duplication of what DOS does for you, let me explain the disk facts of life.

If DOS did what you may think it is supposed to do when you format the disk, DOS would map around these bad areas. Unfortunately, DOS doesn't do this.

DOS 2.0 and 2.1 can't enter the bad tracks. DOS 3.0 can, but only on the IBM AT. Unfortunately, as the press has so well documentated, the AT's hard disk develops bad tracks later on.

We do what DOS can't

We believe the problem is so bad, we use a software program that performs a powerful test of your disk drive on all of the IBM or IBM compatible computers—PCs, XTs, and ATs. Our format takes hours to analyze the disk. But when we finish, you know that the bad tracks are really mapped out so you won't write good data that will disappear into a black hole. We even send you a printed statement of our test results.

Our software allows you to type in the bad track locations from the list supplied by the manufacturers, so you'll never write good data to them—even if DOS didn't identify them as bad. The software even lets you save the location of these bad sections to a file, so that you can reformat your disk without spending hours retesting.

We even include a program that will give you continuous comments on the status of your hard disk. No more waiting for that catastrophic failure.

Average Access Time

As you might suspect, some hard disks are faster than others in their ability to move from one track of data to another. The time it takes the hard disk to move one-half way between the beginning of the disk to the end is called the "average access time."

The first generation of 10 megabyte hard disks had average access times of 80-85 milliseconds (msec). But computer users love speed, and guess what—the average access time for the new 20 megabyte hard disk in the IBM AT is only 40 msec. (We sell an AT equivalent with only 30 msec access time!)

There are some legitimate reasons for the shorter access time. It's particularly helpful when there are multiple users on the same hard disk. It's also important when running a compiler. But remember, before you get too wrapped up in the access speed, there's always that ST 506 interface which won't let data transfer from the hard disk to the computer any faster than 5 megabits/second. We've bypassed that choke hole, too. If you want the functional equivalent of a Ferrari with a turbocharger, order our 10 Mbit per second 100 megabyte hard disk with 18 msec of average access speed.

Compatibility

To be sure that your hard disk is 100 percent compatible with the IBM XT you don't need to buy the same hard disk that's in the XT. You can't even be sure what brand hard disk it is because IBM, like Express Systems, goes into the marketplace and buys hard disks from several vendors. However, they buy their XT hard disk controller from only one vendor—the same one we do.

You can buy the IBM XT controller from IBM for $495 or you can buy from us, the functional equivalent, manufactured by the same company that makes it for IBM for only $195. Is it the exactly identical IBM XT controller? No, it's better. First, it takes less power, and secondly, it can control from 5 to 32 megabytes—the IBM controller can work with only 10 megabytes. It is 100 percent IBM XT compatible, and 100 percent is 100 percent. If you want to save a slot, we carry a version that lets you operate two hard disks and two floppy disk drives.

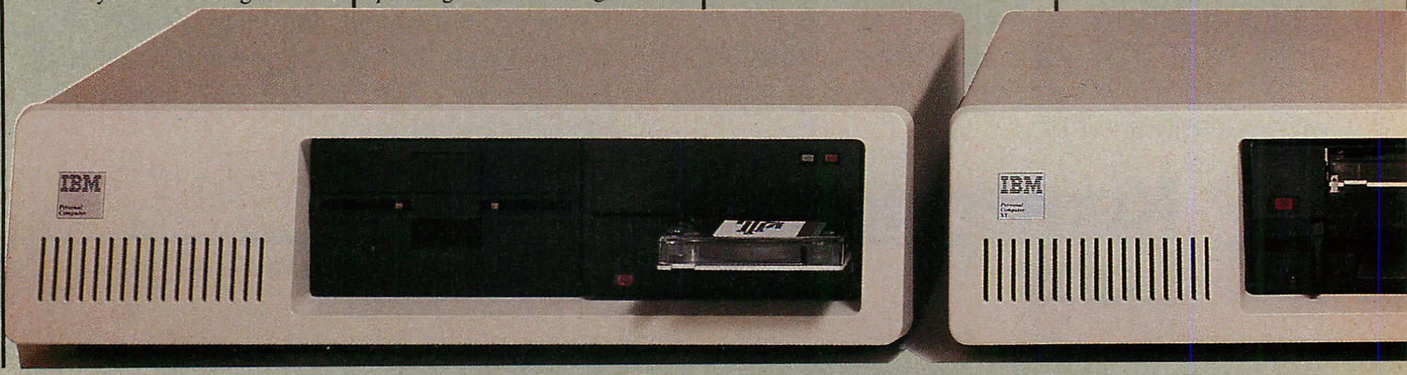
More than 32 Megabytes

You can operate with more than 32 megabytes (the limit of DOS) through the use of "device drivers." Express Systems can supply you with device drivers for our hard disks for over 32 megabytes formatted. But, if you don't have individual files, or databases that are large, you might want to consider one of our controllers that can divide our 65 megabyate (formatted) hard disk into two equal volumes of 32 megabytes each.

Reliability

We offer you a choice between iron oxide and plated media—the stuff that covers the hard disk and gives it its magnetic properties. Iron oxide is,—well, it's rust. If you inadvertently joust your disk, you may cause the low flying head to dig out some iron oxide. A little rust flake can ruin your whole day. Plated media is more resistant to damage, and if it happens, less data is lost.

We offer both types of hard disks. The iron oxide is older



technology, and quite frankly, manufacturers understand it better. Their better understanding, combined with some of the special head locking mechanisms, gives us peace of mind when we sell you one.

Power

Hard disks consume power. Our small, half-high hard disks consume so little power that you can use them with your existing IBM PC power supply. If you plan to use lots of slots, you'll want to increase your power supply to be safe. We offer the same amount of power for your PC that comes in the XT.

Our Customers

Some folks just never feel comfortable buying mail order. They forget that Sears began as a mail order house or that IBM is now into mail order. But, if it helps, here is a *partial* list of customers who have felt comfortable to buy from us.

IBM	Sears
American Express	Honeywell
U.S. Army	MIT
AT&T (Bell Labs)	RCA
Bausch & Lomb	Lockheed
Xerox	Sperry

Easy to Install

If you're like most of us, raised on the boob tube rather than the Great Books, you'd rather see the movie than read the book. Well, now you can choose to read our installation manual or for only $9.95 more, you can get a VHS or Beta video cassette showing the simple steps for installation.

Warranty

We offer you a one year warranty on our hard disks—the same as IBM on the AT and 90 days on the tape drives. (It's all the manufacturer gives us.) If anything goes wrong with your tape or disk drive or hard disk, send it back in the box it came in. However, we have found that we can usually solve the problem over the phone. So call first for a return authorization number because we can't accept any returns without it.

Comes complete

All Express Systems products come complete with the appropriate software, tape and/or hard disk controllers, and cables where required. Hard disks are formatted and tested with the PC DOS of your choice. All drive sizes are formatted capacities.

If your application requires a stacking kit, power splitter cables, daisy chain cable, or some other variation, we'll supply these items at a nominal charge. We even ship our hard disks with Command Assist™ an on-line DOS-like manual to give you help with your DOS commands.

Complete Hard Disk Kits

Formatted MB	Height	Plated Media	Average Access	Transfer Rate	PC or PC/XT	AT
10	½	no	85 msec	5 Mbits/s	$ 395	$ N/A
10	½	yes	85 msec	5 Mbits/s	$ 495	$ N/A
21	½	yes	85 msec	5 Mbits/s	$ 795	$ 595
21	Full	no	30 msec	5 Mbits/s	$ 1,535	$ 1,340
32	½	yes	85 msec	5 Mbits/s	$ 995	$ 795
32	Full	no	30 msec	5 Mbits/s	$ 1,775	$ 1,575
65	Full	no	30 msec	5 Mbits/s	$ 2,295	$ 2,070
100	Full	yes	18 msec	10 Mbits/s	$ 4,995	$ 4,995
Removable Hard Disk						
10	½	no	90 msec	5 Mbits/s	$ 1,095	N/A

Tape Systems and Subsystems

Formatted Storage Capacity	Height	Data Transfer Rate (k/sec)	PC or PC/XT	AT
60 Mbytes	½	88	$ 995	$ 995
60 Mbytes Subsystem		88	$ 1,295	$ 1,295
21 Mbytes (unformatted) Start/stop Subsystem		24	$ 595	$ 595
26 Mbytes Floppy Tape® Subsystem		31	$ 749	$ 749

Controllers

All of our hard disk and tape controllers are available separately: Please call for prices.

Subsystem Chassis

Any of our disk or tape units are available in an external subsystem for an additional $250.00. You can mix & match any of our ½ high hard disks or tape drives together or add any single full height hard disk.

Tape Cartridges

Express Certified 555 foot 310 Hci ¼-inch Data Cartridge	$35.00

Power Supply

130 Watt Power supply	$75.00*
150 Watt Power supply	$125.00

*with the purchase of any drive

More questions?

Because we spend so much attention on the front end with ensuring that our disks will arrive in working order, we have a customer service department that, unlike many of our competitors, has little to do. When you need us, you won't get a constant busy signal.

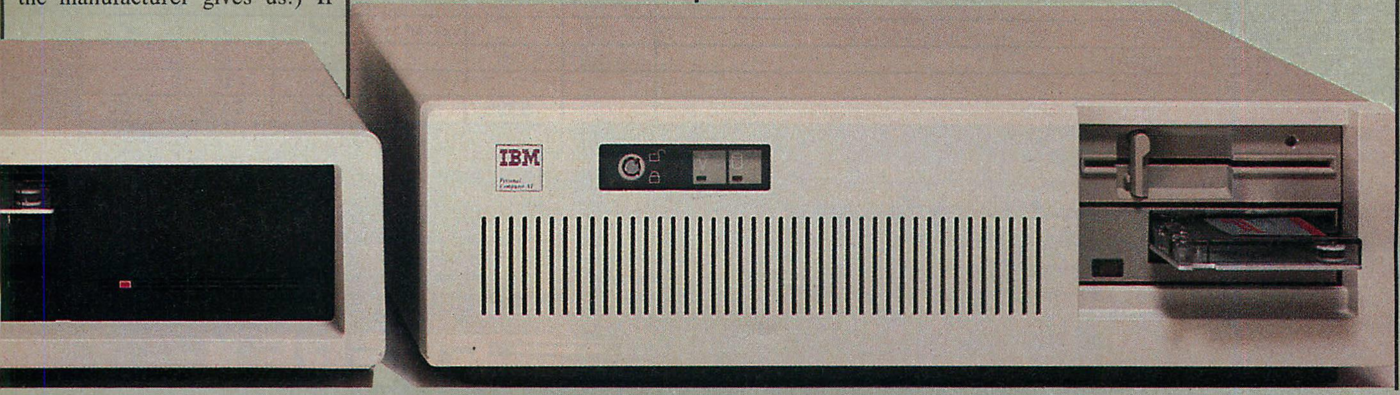
Call our friendly, knowledgable customer service staff to get answers to your questions—before or after the sale. Our people, who know the PC, can talk you through the sticky parts, and they'll respond to you quickly. Just call us.

How to order

Pick up the telephone and call 1-800-341-7549, to order. We accept Master Card, VISA, American Express and Diners Club. Or send a cashier's check or money order (We'll take a check, but you'll have to wait for it to clear) and tell us if you want one of our recommended configurations or you want to mix and match yourself. Corporations with a DUNS number may send purchase orders for quantities over five.

EXPRESS SYSTEMS

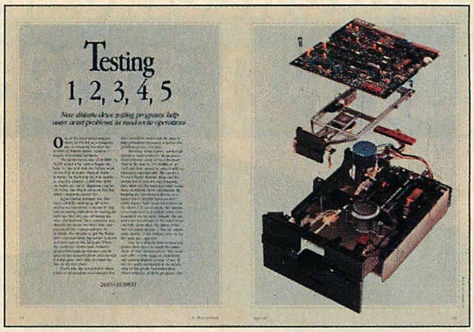
Call Toll Free 1-800-341-7549 Ext. 1300
In Illinois call (312) 882-7733 Ext. 1300
Express Systems, Inc., 1254 Remington, Schaumburg, IL 60195



Command Assist is a trademark of Micro Design International

CIRCLE NO. 155 ON READER SERVICE CARD


Graphic Enhancement


Testing 1, 2, 3, 4, 5

GRAPHIC MISCONCEPTIONS

The article, "Graphic Enhancement" (Thomas V. Hoffmann, April 1985, p. 58), contained an error, stating "The VDI support for EGA does not take full advantage of the colors available. . ." This is not true. The VDI gives complete access to all EGA graphics capabilities, including color, in a device-independent way. The color capabilities of the EGA are supported via VDI's SET COLOR REPRESENTATION function. A C programmer, for example, would include the following call and parameters in his source code:

```
vs_color (handle, index_in, RGB_in,
  RGB_out)
```

Through this call the Graphic Software Systems (GSS) VDI allows the programmer to set an index to an arbitrary RGB color; the EGA device driver then returns information to tell him which specific EGA RGB color was selected (VDI's device independence means that it maps a generic request into the EGA's capabilities).

An important benefit of VDI is its support of the EGA color map. This allows programmers to take advantage of all EGA color capabilities, while still maintaining the portability and device independence of the application.

PC programmers are hungry for information on graphics applications development, and the lack of widely accessible information on the Virtual Device Interface (VDI) has led to several popular misconceptions.

Performance degradation. The GSS VDI should cause no more than 10 to 15 percent overhead compared to writing directly to the hardware; the trick is knowing how to use it properly. GSS benchmarks also indicate that the EGA drivers are optimized well enough to run as fast as our newly available optimized Color Graphics Adapter (CGA) drivers (version 1.03).

Availability of drivers. GSS markets a library supporting 46 devices beyond those supported by IBM's Graphics Development Toolkit (the three volumes are available for $39.95 each.

Peripheral capabilities. The drivers support the full graphics capabilities of their devices—and go beyond that—because the VDI emulates capabilities not offered by the devices.

Recompilation. New drivers (new peripheral support) is added to VDI-based applications without modifying or recompiling the application. Program recompilation is necessary only to go from one CPU/OS environment to another (for example, from the IBM PC to the UNIX PC 7300, but not from the PC to a PC clone with a different screen).

GSS is developing new VDI capabilities, new drivers, and new tools. IBM is always moving ahead with new hardware and system software capabilities.

Daniel Fineberg
Graphic Software Systems
Wilsonville, OR

Mr. Fineberg is correct. The VDI default settings define only eight distinct colors for indexes 0 to 7, with indexes 8 to 15 all set to white, but these defaults can be changed as Mr. Fineberg describes.
—Thomas Hoffmann

PC Tech Journal *offers a thorough explanation of VDI in this issue. See "Forward-Looking VDI," Steven Armbrust and Ted Forgeron, p. 42.*
—WF

MANUAL SEARCH

In the May 1985 article, "Testing 1,2,3,4,5" (p. 142), James Chumbley remarks that "a competent (and determined) user" who wants to adjust a disk drive "must have a copy of the drive's factory service manual in hand." I wholeheartedly agree.

My drives are the standard IBM version of the Tandon 1002, and I have a spare Tandon on hand, just in case. Competence I cannot speak for, but determination I can. Now, where can I get the factory service manual? I can find sources for the Tandon drives, but not for the manual. IBM will sell at a great price a service manual for the entire PC, but I don't have the great price and I don't want the whole thing. No dealer I have asked has a clue as to how to obtain a Tandon manual.

So I turn to you. Can you supply a source for these manuals?

Richard N. Wisan
Oneonta, NY

Our research revealed that a manual is available for $29.50 (including shipping and handling) from the Tandon Corp., P.O. Box 2107, Chatsworth, CA 91311. Telephone: 818/701-4621.
—WF

DISKS OF ALL SIZES

The article, "The Third Drive" (Jack Wright and David Zarodnansky, June 1985, p. 47) gave an excellent description of how to connect external floppy disk drives to the IBM PC. It is a fairly inexpensive way to increase storage.

The authors did not mention that it is possible to connect drives for other types of media through the external connector. In particular, many of the microfloppy (3½-inch) disk drives are electrically compatible with the 5¼-inch drives. The disks can be formatted as normal PC-DOS 360KB disks, or by using a custom driver they can access 720KB or more on a double-sided disk. With the proper format, disks from non-compatible computers, such as the Data General One or Hewlett-Packard 150, can be accessed by the PC.

Because the disk is formatted as a standard PC-DOS block device, all PC-DOS software will run on the microfloppy just as on a 5¼-inch disk.

These durable high-capacity diskettes are great for backing up hard disks or archiving data. The conve-

nience of carrying the small disks in your pocket is an added bonus.

Steve Potocny
Compufirm Corp.
San Diego, CA

ESCAPE TO COMPATIBILITY

I much enjoyed Ted Forgeron's article on "Machine Specifics" (June 1985, p. 181). The subject of portability of software can hardly be overemphasized. However, I find it ironic the program, MATHCHIP.ASM, in listing 4 of that article is unnecessarily incompatible with release 1.0 of the IBM Macro Assembler.

Because the mnemonics, **fninit** and **fnstcw**, assemble as escape sequences, they may be substituted in the assembly listing and assemble the source with any release of the Macro Assembler. Thus, changing the lines

```
fninit              ;initialize math chip
```

to

```
esc 1ch,bx          ;initialize math chip
```

and

```
fnstcw control      ;store control word
```

to

```
esc 0fh,control     ;store control word
```

in the MATHCHIP.ASM listing will produce a source that can be assembled with the IBM Macro Assembler 1.0.

In general, all 8087/80287 mnemonics may be expanded into Esc sequences, although anyone who makes heavy use of these instructions would probably wish to code a set of assembly macros to do the translations automatically or purchase a newer version of the Macro Assembler.

James L. Stadelmann
Chicago, IL

Mr. Stadelmann is absolutely correct. The Esc instruction provides the mechanism by which coprocessors receive their instructions. Recognizing that numeric coprocessor instructions assemble into escape sequences could save the cost of upgrading from MASM 1.0 to a more recent version of the assembler. As for me, I bought an upgrade. Frankly, I'm a sucker for new versions of software.
—Ted Forgeron

TURBO MODS

Tech Notebook 40 ("Saving Space in Turbo," John Figueras, June 1985, p. 39) was a good idea, especially for users who are not blessed with hard disks.

I have modified Mr. Figueras' program slightly to make it even more useful to those who have Turbo Pascal 3.0.

Turbo 3.0 offers two handy functions called ParamCount and ParamStr. They are analogous to the **argc** and **argv** variables in C. ParamCount returns the number of parameters that were typed on the command line, and ParamStr (N) returns the *N*th parameter as a string. These can be used to eliminate the intermediate 'Run which Program' prompt in Mr. Figueras' driver program.

By the way, I think the *PC Tech Journal* BBS is a great idea. If it had been up when I wrote this, I would have sent it online, except that my modem is old (read 300 baud).

Lionel Gibbons
Montreal, Quebec

We have slightly modified your modification and printed it here. Thank you for your contribution.

Please note that PCTECHline does accept 300-baud modems; however, all callers are disconnected after one hour.
—WF

```
program run;

{activate chain file}

VAR
  DiskFile : file;

BEGIN
  IF ParamCount < 1 THEN
    writeln(chr(7),'USAGE:   A>RUN <filename>')
    { Note: This area could also be used to
      implement the menu-driven option
      described by Mr. Figueras. }
  ELSE
    BEGIN
      assign(DiskFile,ParamStr(1) + '.CHN');
      chain(DiskFile)
    END
END.
```

BASIC BIAS

The editor's response to George Dinwiddie ("Another CRC Routine," Letters, July 1985, p. 16) about his CRC algorithm becoming "intolerably slow if implemented in a high-level language" seems to be BASIC-chauvinistic.

Both the table look-up and bit-manipulation techniques for generating a CRC are well-known. One reference to the table look-up technique is "Byte-wise CRC Calculations" (A. Perez, *IEEE Micro*, June 1983). A number of references illustrate bit-manipulation techniques from at least 1976 (see "Calculating and Error Checking Character in Software," S. Vasa, *Computer Design*, May 1976), and the same idea was presented in hardware articles starting at least three years earlier. The clever use of the Intel parity flag is a good trick

for CRC-16, but unfortunately it does not work for CRC-CCITT.

The bit manipulation technique will usually work well when implemented in C, or, more recently, in Turbo Pascal. A Turbo procedure can compute a full byte-CRC (CRCITT) in 389 microseconds on an 8088 with 4.77 mHz clock) *without* using in-line machine code. Since this is less than 40 percent of a 9600-bps character time, the bit manipulation technique does not seem to me to be "intolerably slow" in a high-level language.

The bit-manipulation technique may be difficult in BASIC, and in that case the table look-up approach (with its associated 512-byte table of constants) is probably justified. But the real problem may be in using BASIC instead of C or Turbo. Does BASIC still qualify for the term "high-level?"

Terry Ritter
Austin, TX

;COMMENT "OMISSION ERROR

As a novice in the computer business, I followed your instructions in Tech Notebook 41 ("Keyboard Scan Codes," Ted Mirecki, June 1985, p. 40) and entered the program exactly as printed in *PC Tech Journal*. Upon assembly with my IBM MASM version 2.0, I was rewarded with ten errors. Closer inspection of your listing reveals that the typesetter has forgotten to put semicolons in front of the comments. Upon inserting the semicolons, the program assembles and runs correctly.

Douglas S. Egan, Jr.
Amtrak
Philadelphia, PA

The omission was not semicolons, but the second set of double quote marks, another way that comments can be set off in assembly language. Either way, we were at fault, and we apologize for the inconvenience.

—WF

TRUE OR FALSE?

I have read the July 1985 issue of *PC Tech Journal*, and, as a user of Turbo Pascal, was interested in Tech Notebook 44 (Tom Swan, "Comparing Structures," p. 36). Unfortunately, the program, however, fails in a subtle but catastrophic way.

In Turbo Pascal, as in other languages that use variable length strings, two strings are considered equal if the characters in their significant lengths are equal. If the significant length of a string is less than its declared maximum

length, any characters beyond the significant length are of no consequence and are not considered in any tests or manipulation of the string.

In compare.pas, the structured variables are treated as arrays of byte. If the record structure of the variables contains strings, every byte of these strings will be compared, including those bytes between the significant length and the maximum length. These characters can be different without affecting the equality of the strings. This possibility can best be demonstrated by making a change to the main block of the program as shown below.

If you run this version of the program you will be told that r1 and r3 are

```
BEGIN
  r1.name := 'R. Reagan'; r1.age := 74;
    r1.elected := 1980;
  r2.name := 'J. Carter'; r2.age := 61;
    r2.elected := 1976;
  r3 := r1;
  r3.name := 'G. Washington';
  r3.name := 'R. Reagan';
  comparerecs(r1,r2);
  comparerecs(r2,r3);
  comparerecs(r1,r3);
END.
```

not equal. This is because the comparison is being done not between the two strings r1.name and r3.name, which are equal, but between two arrays of bytes which contain the following characters: r1 'R. Reagan ' and r3 'R. Reagangton '.

Tom Swan's routine will work fine if the structured variables do not contain strings, but if they do, then its back to field-by-field comparisons. This same reasoning probably explains why Borland does not support the comparison of structured variables in Turbo Pascal.

Peter W. Roach
W. Springfield, VA

PC Tech Journal is the best popular source of information for IBM PC users. A feature I especially like is the Tech Notebook series of programming tips. In the July 1985 issue, Tech Notebook 44 describes a Turbo Pascal routine of comparing records for equality by bitwise checking their memory images. This method does not work

The reason is that string assignments in Turbo do not reinitialize the unused space reserved by the compiler in the string's declaration. If the records compared contain strings, Mr. Swan's Equalrecs may return False when the records are in fact identical. A solution would require using more information about the record definition than one would normally want to give an all-purpose routine.

There is another problem with Mr. Swan's Equalrecs. Since he uses Rec in the routing, a different version would be necessary for every type of record to be compared. However, because Turbo allows the user to pass "typeless" variables to routines, the function Compare in the following example does what Equalrecs does for any pair of contiguous-memory arrays of equal length.

Scott Ferson
Stony Brook, NY

```
FUNCTION
Compare (var A,B; Size : integer) : boolean;
{True only if Size bytes staring at A are the
same as those starting at B}
TYPE image=array[1..maxint] of byte;
VAR c : image absolute A;
    d : image absolute B;
    i : integer;
BEGIN
compare := true;
for i := 1 to Size do if c[i] <> d[i]
then compare := false
END;
```

I want to thank the readers for pointing out problems in Tech Notebook 44. I can think of two solutions to the problem with the CompareRecs procedure.

One, fill all record variables with a known value before assigning new values to fields. To accomplish this, use the FillChar procedure as follows:

FillChar(R1, sizeof(R1), 0);

This sets all bytes in record R1 to zeros. Note that this also nulls all string fields by setting their length bytes to 0. Future individual assignments to string fields, however, could cause the bug to reappear. Another possibility is a procedure to make string assignments, then fill the unused characters in a string with blanks, as shown below:

```
PROCEDURE
  StrAssign(Source : BigString;
            VAR Destination : BigString;
            DestSize : integer );
        VAR
          i : integer;
        BEGIN
          Destination := Source;
          FOR i := length( Source ) + 1
          TO DestSize - 1 DO
               Destination[i} := ''
               { single blank char }
        END;
```

Range checking must be off, as it normally is in Turbo Pascal, for the procedure to operate correctly. BigString is a new type declared as String[255], the maximum string length in Turbo Pascal. The procedure can be called in the following way:

{$V-} StrAssign('R. Reagan', R1.name, Sizeof(R1.name)); {$V+}

The {$V-} option tells the compiler to accept string parameters of a smaller size than expected by the procedure. By inserting the above procedure and making all string assignments as described here, CompareRecs operates correctly. Using Mr. Roach's example, the new program is as shown below:

```
BEGIN
  {$V-}
  StrAssign
    ( 'R. Reagan', r1.name, sizeof( r1.name ));
     r1.age := 74; r1.elected := 1980;
  StrAssign
    ( 'J. Carter', r2.name, sizeof( r2.name ));
      r2.age := 61; r2.elected := 1976;
  r3 := r1;
  StrAssign
    ( 'G. Washington', r3.name, sizeof( r3.name));
  StrAssign
    ( 'R. Reagan', r3.name, sizeof( r3.name ));
  comparerecs( r1, r2);
  comparerecs( r2, r3 );
  comparerecs( r1, r3)
END.
```

Despite the potential bug, I would hate to give up direct record variable comparisons, and hope that these ideas help readers make use of CompareRecs. Thank you for giving me the opportunity to straighten this out.

—Tom Swan

SEARCHING FOR MULTIUSER DOS

Does there exist a true multiuser MS-DOS operating system? (See "Concurrent PC-DOS," Don Awalt, March 1985, p. 45.) I have been searching for one everywhere. I find it hard to believe that a sophisticated, well-marketed machine does not have such an operating system available. I have lately worked on the Altos 580 with MP/M, a CP/M version of a multiuser operating system. It did have limitations, but it was truly a multiuser system.

Susan F. Scarborough
Mountain View, CA

The MP/M operating system was developed during the pre-IBM PC era when computers were able to support multiple terminals, lots of memory and disk space, and reasonable CPU speed.

The IBM PC consisted of single-user/single-tasking hardware, operating

system, and applications software. Even the availability of MP/M for the IBM PC was inappropriate, due to insufficient multiuser applications performance.

Although no true multiuser versions of MS-DOS exist, progress continues. IBM is enhancing the power of its PC family; the AT is much closer to a multiuser machine than the PC. Microsoft regularly upgrades MS-DOS; network and multitasking features appear in MS-DOS 3.1 and Windows, respectively, with Windows providing evolution to a future MS-DOS 4.0. Finally, applications software is being upgraded in order to include network functionality under MS-DOS.

The mysteries of a multiuser IBM PC will be discovered soon.

—Don Awalt

WORK-AROUNDS

As a user of Microsoft Pascal 3.13, I was very interested in Ted Forgeron's article on "Pascal Bugs" (May 1985, p. 199). Bug number 3 involving parameters to library functions especially concerned me, because I use these functions extensively. I was, however, able to find several ways around this bug to enable the use of the routines without the need for passing the variables as VAR parameters.

First, it should be noted that the library functions work properly if the parameters are global variables. I tested this with Forgeron's routine by using the global variable *x* as the parameter within the **pass_var** routine.

Second, for those working with REALB variables, the Pascal library supplied with the Microsoft 3.13 includes extended functions that force a particular precision. When using these functions the errors associated with VAR parameters disappeared. Granted, use of these library functions can add to the program size, but they could be used in instances where the programmer does not wish to pass the complete segmented address of the variables.

Being a faithful user of Microsoft compilers, I would never have questioned the supplied library functions if not for Ted Forgeron's excellent article.

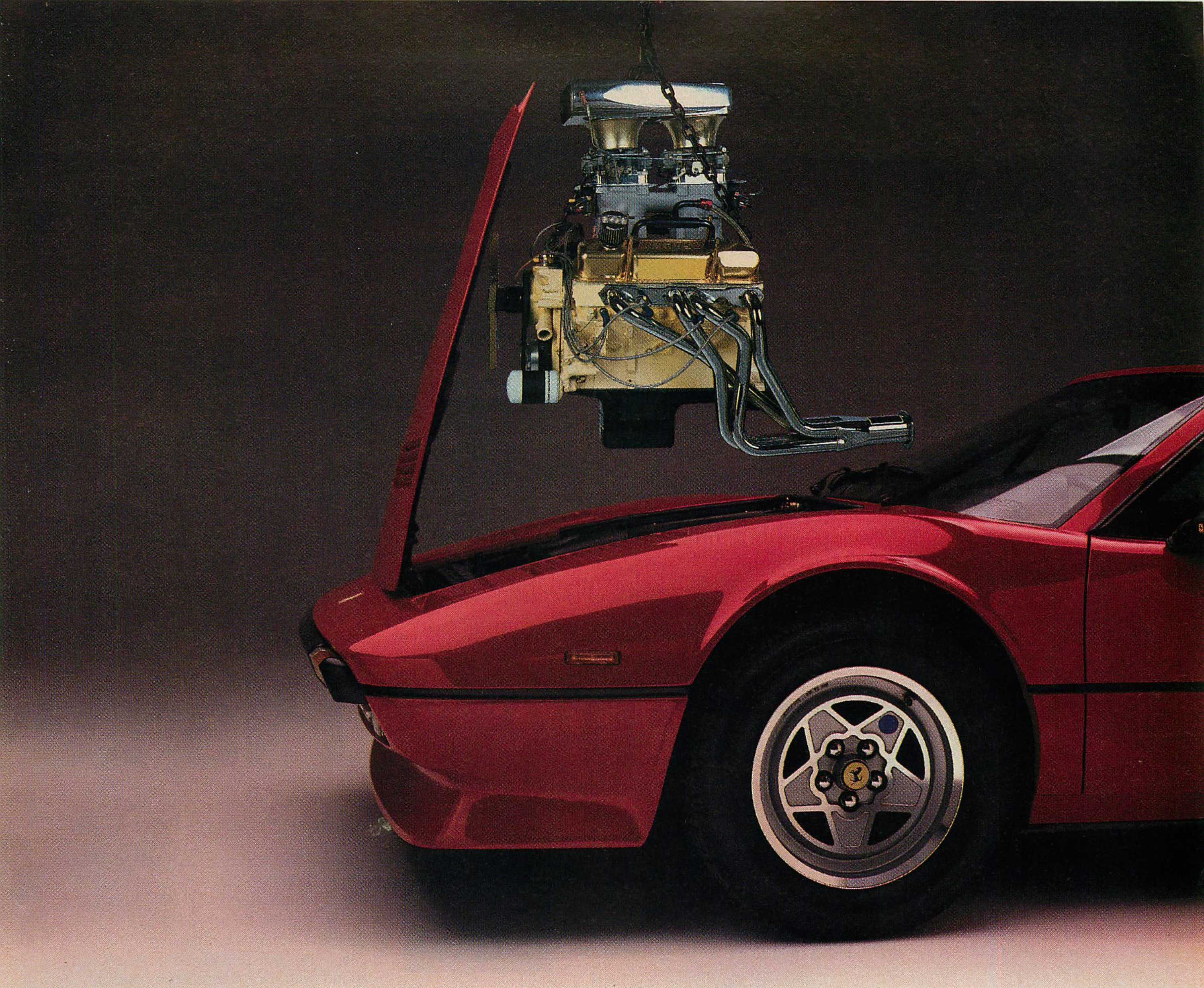
Gregory L. Hendricks
Norris, TN

Few people know about Microsoft's predeclared arithmetic functions. For example, a single-precision (REAL4) cosine can be calculated with a call to CNSRQQ. Likewise, a double-precision (REAL8) cosine is calculated by CNDRQQ. Precision-specific functions also exist for ABS, ARCTAN, EXP, LN, SIN, SQR, and SQRT. Fortunately, bug number 3 is fixed in MS-Pascal 3.3.

—Ted Forgeron

ERRATUM

Some of the PC/IX benchmarks for the PC/XT were incorrectly listed in table 3 of the "PC/IX" article (Augie Hansen, July 1985, p. 80). The correct figures for compile time (real) are: empty, 0:25; sieve, 0:29; hello, 0:28; for execution time (real): empty, 0.3; sieve, 7.1; hello, 0.6; for execution time (user): empty, 0.0; sieve, 6.3; hello, 0.0; for execution time (system): empty, 0.3; sieve, 0.3; hello, 0.4; for foreground sequential (real): make sortdemo, 0:54; make xsort, 0:38; total elapsed time: 1:32; for background/foreground: (make sortdemo; echo "\07")&, 1:51; make xsort, 1:37; total elapsed time, 1:51; for background "simultaneous": (make sortdemo; echo "\07 1")&, 1:58; (make xsort; echo "\07 2")&, 1:36; total elapsed time, 1:58.



AT™ Pfantasies for your PC or XT.™

Want better speed and memory on your PC or XT without buying an AT?

You've got it!

Phoenix's new Pfaster™286 co-processor board turns your PC or XT into a high-speed engine 60 percent faster than an AT. Three times faster than an XT. It even supports PCs with third-party hard disks. But that's only the beginning.

You can handle spreadsheets and programs you never thought possible. Set up RAM disks in both 8088 and 80286 memory for linkage editor overlays or super-high-speed disk caching. All with Pfaster286's 1mb of standard RAM, expandable to 2mb, and dual-mode design.

You can develop 8086/186/286 software on your XT faster. Execute 95 percent of the application packages that run on the AT, excluding those that require fancy I/O capabilities your PC or XT hardware just isn't designed to handle. Queue multi-copy, multi-format print jobs for spooling. Or, switch to native 8088 mode to handle hardware-dependent programs and back again without rebooting. All with Pfaster286's compatible ROM software.

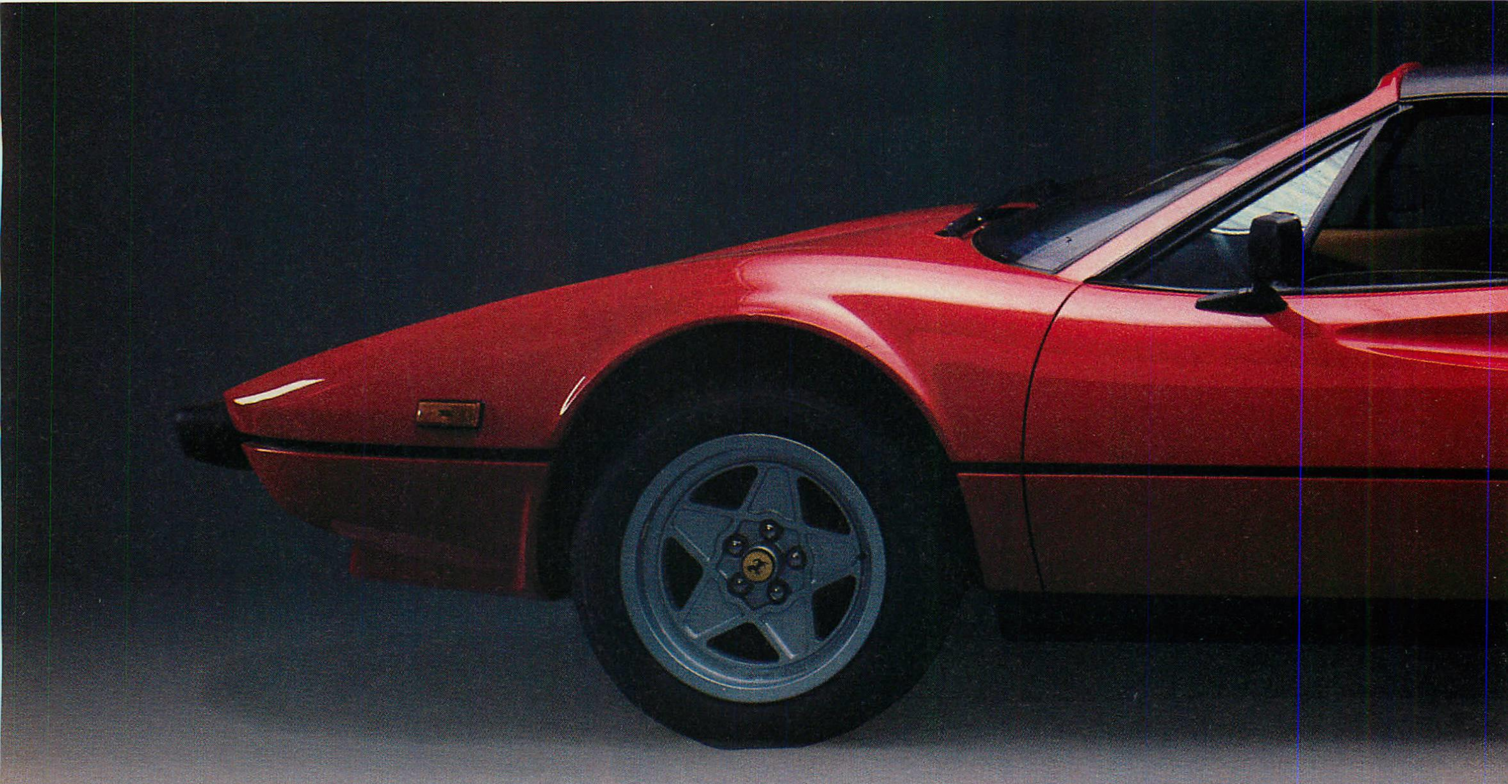
And, Pfaster286 does the job unintrusively! No motherboard to exchange. No wires to solder. No chips to pull. Just plug it into a standard card slot, and type the magic word, "PFAST."

If you really didn't want an AT in the first place, just what it could do for you, call or write: Phoenix Computer Products Corp., 1416 Providence Highway, Suite 115, Norwood, MA 02062; (800) 344-7200. In Massachusetts, 617-762-5030.

Programmers' Pfantasies™
by

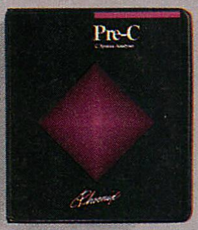
Phoenix

XT and AT are trademarks of International Business Machines Corporation. Pfaster286 and Programmers' Pfantasies are trademarks of Phoenix Computer Products Corporation.
For the Ferrari aficionado: yes, we know this is a rear engine car. We are showing the addition of a second engine to symbolize how Pfaster can be added to your PC or XT to increase performance.



Programmers' Pfantas

Phoenix makes programmers' dreams come true. With the best-engineered, highest performance programming tools you can find. A full line of MS-DOS©/PC DOS programs and utilities no other company offers. All designed to help you write, test and deliver the best programs possible. Top-of-the-line quality at a price you can afford. And all these products are available for the IBM© PC, XT,™ AT™ and compatibles.



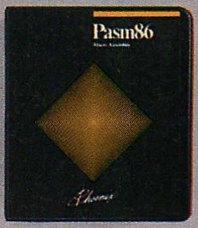
Finally, A Lint For MS-DOS.

Now you can get the full range of features C programmers working in UNIX™ have come to expect from their Lint program analyzer.

With Pre-C™ you can detect structural errors in C programs five times faster than you can with a debugger. Find usage errors almost impossible to detect with a compiler. Cross-check multiple source files and parameters passed to functions. Uncover interface bugs that are difficult to isolate. All in a single pass. Capabilities no C compiler, with or without program analyzing utilities, can offer. In fact, Pre-C outlints Lint, since you can handle analyses incrementally.

Pre-C's flexible library approach lets you maintain continuity across all the programs in your shop, whether you use Pre-C's pre-built libraries, pre-existing functions you already have, or some you might want to buy yourself.

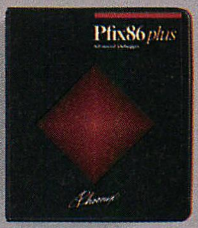
Plus, you're not limited to one particular library, and Pre-C keeps track of all the libraries you're using to make sure that code calls them correctly. **$395.**



Assemble Programs Twice As Fast.

Pasm™86 will assemble MASM files two to three times faster than MASM 3.0. Pasm86 supports 8086/88, 8087, 80186 and 80286 processors.

With Pasm86's built-in defaults, you can write code quickly since you won't spend hours learning all the control statements needed at the beginning of your program. You can define symbols on the command line. Decide whether you want error messages or not. And, put local symbols within procedures. **$295.**



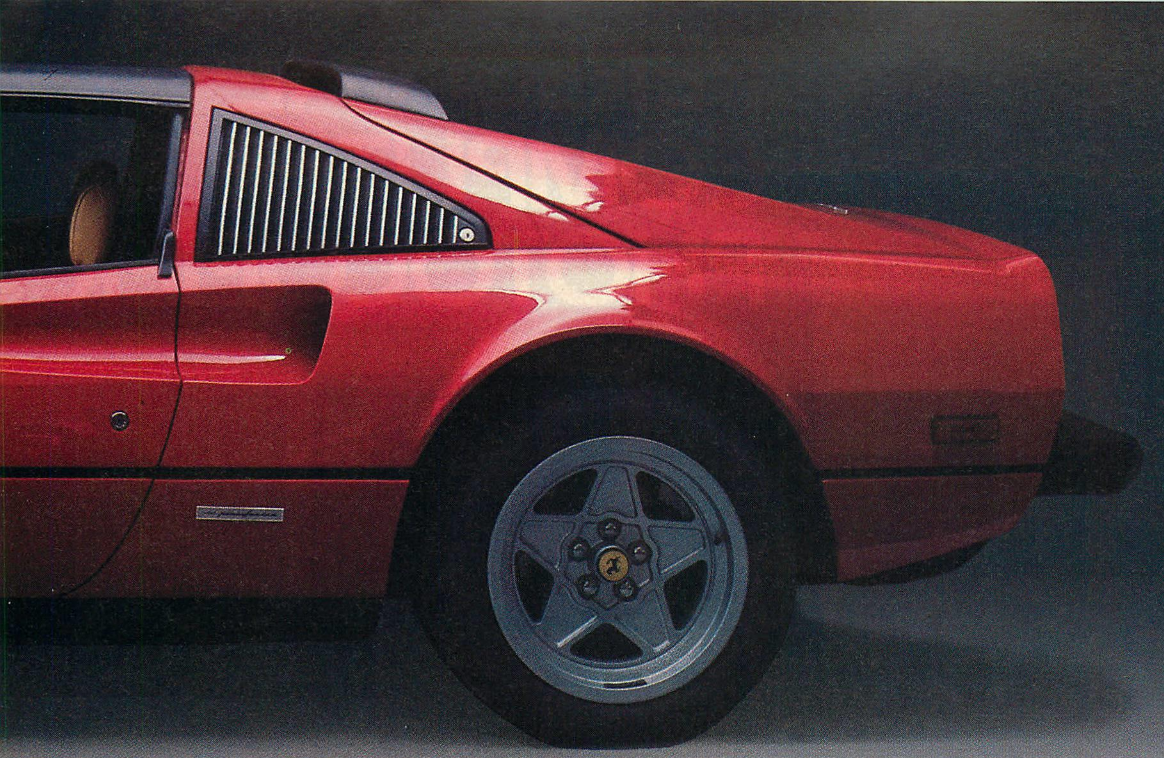
Still Fixing Bugs The Hard Way?

Pfix™86 Plus, the most advanced symbolic debugger on the market, eliminates the endless error searches through piles of listings. Locate instructions and data by symbolic name, using symbolic addresses. Handle larger, overlayed programs with ease.

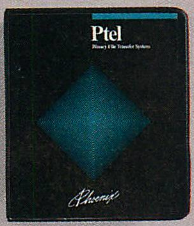
An adjustable multiple-window display shows source, object code and data, breakpoint settings, current machine register and stack contents simultaneously. An in-line assembler allows program corrections directly in Assembly language. Powerful breakpoint features run a program full speed until a loop has been performed n times.

With a single keystroke you can trace an instruction and the action will be immediately reflected in source, object, data, stack, and register windows. Another key begins a special trace mode that executes call and loop instructions at full speed. Designed to work with both Plink™86 and MS© LINK linkage editors. **$395.**

Outside the US, contact: Lifeboat Japan, Tokyo, Japan, Telex #2423296 LBJTYO, Telephone: 03-456-4101 • Repro Haganum (Stevis BV), Dinther, Netherlands, Telex #39581 REHAGNL, Telephone: 01-720-74543, 01-720-75543 • Memory Data, Sundbyberg, Sweden, Telephone: 46-8764-6700 • Roundhill Computer Systems, Marlborough, Wiltshire, UK, Telex #444453 AWARE G, Telephone: 44-467254675.



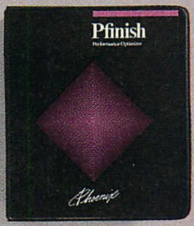
ies™ by Phoenix



Get The Lead Out Of Binary File Transfer.

Ptel™ is the universal binary file transfer program for MS-DOS 2.0 or higher. You can move binary files fast and accurately. Upload or download groups of files from Bulletin Boards or remote computers. Move files between dissimilar machines and operating systems. Ptel's advanced binary protocol, Telink,™ offers better-than- Modem7 accuracy and performance. Faster transfer speeds. An on-screen update of error correction, blocks transferred, and transmission time.

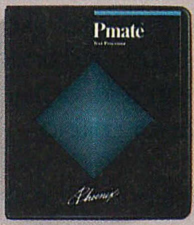
Includes popular Modem7 and XModem protocols. With checksum or CRC. Plus Kermit and ASCII. **$195.**



Maximize Your Program's Efficiency.

Pfinish™ delivers the fastest running programs possible. This performance analyzer lets you ''zoom in'' on the inefficient parts of your program. Whether written in Assembly language. C. PASCAL. FORTRAN. BASIC. Unlike profilers available today, Pfinish understands the structure of your program and reports the amount of activity and time spent in its subroutines or functional groups. Pfinish analyzes both overlaid and memory resident programs. Down to the instruction level. Reports are displayed. Stored on disk. Or printed out. In tabular form or histograms.

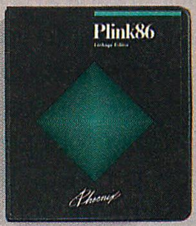
Do a dynamic program scan. Identify the most frequently executed subroutines. Find inefficient code that costs your program valuable time. Rank subroutines by execution frequency. **$395.**



Why Work With A Primitive Editor?

More than a powerful editor, Pmate™ is a text processing language. An emulator of other editors. A language-specific editor for C, PASCAL, and FORTRAN. Pmate can even run in the background!

You get full-screen, single-key editing. Ten editing buffers. Horizontal and vertical scrolling. A ''garbage stack'' buffer. A built-in macro language with variables, control statements, radix conversion, tracing and 120 commands that you can group and execute with a single keystroke. **$225.**



Why Squeeze Your Program More Than You Have To?

The Plink86 overlay linkage editor brings modular programming to 8086/88-based micros. Write large and complex programs without worrying about memory constraints. Work on modules individually, link them into executable files. Use the same module in different programs. Change the overlay structure of an existing program without recompiling. Use one overlay to access code and data in other overlays.

Plink86 links Intel®-format modules. **$395.**

Call (800) 344-7200. In Massachusetts (617) 762-5030. Or, write.

Phoenix

Phoenix Computer Products Corp.
1420 Providence Highway Suite 115
Norwood, MA 02062

Intel is a registered trademark of Intel Corporation.
XT and AT are trademarks of International Business Machines Corporation.
UNIX is a trademark of AT&T Bell Laboratories.
Pasm86, Pfix86 Plus, Ptel, Telink, Pmate and Plink86 are trademarks of Phoenix Software Associates Ltd.

CIRCLE NO. 205 ON READER SERVICE CARD

PERSONAL COMPUTER PHOTOCOPIER

Have you ever wished that there was a machine for your PC that could scan your photographs, artwork or documents just like a photocopier? And was as easy to use and understand? As well as copy and store in color or black-and-white onto your hard drive for editing?

Well, there is such a machine now. It's called the **LaserFAX** digital photocopier. We got tired of trying to use digitizing tablets, cameras, mice and everything else that has been invented. The **LaserFAX** machines will scan any and all of your art, photos, forms or text for storage in your PC. Images that you can cut-and-paste into your word processing or database programs. Our **SpectraSCAN 200** will scan and store your color images for editing and printing on your color printer at resolutions up to 200 dots per inch. The **LaserFAX DS-200** will do the same for you in black-and-white. Then the **LaserFAX Graphics Editor** software that we bundle with the scanner will allow you to cut-and-paste your images, rotate them, enlarge and reduce whatever pictures you choose.

That's not all. Our digital photocopiers are based on open-architecture design, which means that as new uses for this technology develop, add-on cards can enhance our **LaserFAX** digital photocopier. But we didn't want to wait for third-party vendors, so we invented two optional boards of our own: the **LaserFAXimile Card** and the **LaserFAX TEXreader**. The **LaserFAXimile card** takes advantage of the 200 dots per inch resolution of the scanning device to turn it into a facsimile machine. The **TEXreader** board is a revolution in itself — full-scale OCR (optical character recognition) to read typewritten pages with the **LaserFAX** machine into your word processor without having to re-type what somebody else has already typed.

What's most amazing about the **LaserFAX** product line, though, are the prices. Our basic black-and-white digital photocopiers cost about the same as a regular office copier (which you may never use again!) and our color scanners cost very little more. Because we know high tech has to be affordable.

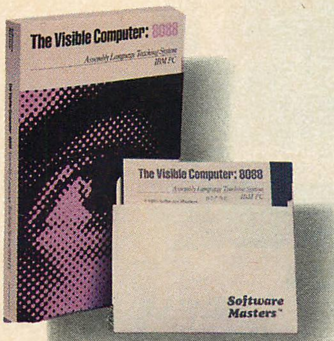
LaserFAX Inc.

2000 Palm Street S., • Naples, Florida 33962

(813) 775-2737

• IBM is a registered trademark of International Business Machines, Corp.
• LaserFAX, SpectraSCAN 200, DS-200, LaserFAX Graphics Editor, LaserFAXimile, TEX reader are trademarks of LaserFAX, Inc.

CIRCLE NO. 194 ON READER SERVICE CARD



Visible Computer: 8088

An understanding of the 8088, through a visual representation, helps the user to learn 8088 assembly language.

Learning a subtle and abstract subject is made easier by choosing a common metaphor and then visualizing the subject in terms of that metaphor. Colliding gas molecules are not at all like billiard balls, but thinking of them in that way can help in understanding what happens when gas molecules collide with each other.

Certainly 8088 assembly language is not as abstract as gas thermodynamics, but it is difficult to learn because it consists of changing binary numbers inside a handful of integrated circuits, where nothing can be seen. A good learning tool for this language must translate those "invisible" numbers into a visual metaphor for the 8088, slow down its operation, and allow it to be single-stepped. Such a tool must include a complete and clearly written tutorial to explain what is happening on the screen. A tall order, but one that is filled by Software Masters' *The Visible Computer: 8088*, earning it *PC Tech Journal*'s recognition as the September Product of the Month.

The Visible Computer: 8088 is an inexpensive book/diskette package designed to be a comprehensive course in 8088 assembly language for the IBM PC. Ambitious in scope, it attempts to take a user from complete ignorance of digital logic all the way to writing interrupt programs in assembly language. Because many programmers come to assembly language by way of BASIC or Pascal and already have some knowledge of computer fundamentals, it is difficult to say how well it achieves that goal. Without question it can teach 8088 assembly language to a computer-literate person extraordinarily well.

TVC's software includes an interactive full-screen debugger that is similar to Morgan's Trace86 or Codesmith-86 by Visual Age. It disassembles code files and displays the contents of registers after each instruction is executed. The debugger, however, is only the lowermost layer of a full 8088 *simulator* that breaks down each individual 8088 instruction into minute steps and explains each step as it happens.

TVC displays all 22 of the 8088 registers, including those not directly accessible to the programmer. It also displays the address bus, data bus, and five 8088 control pins as part of the visual metaphor for the 8088. The user can watch TVC perform a move immediate to register instruction (for example, MOV BX,0FF61H) and see the 8088 fetch the operation code, send addresses out to memory, pick up bytes from memory, move those bytes from temporary registers to the destination registers, and finally, clear prefixes and check the status of the five control pins for pending interrupt and other hardware service requests. TVC offers four levels of simulation; at the level of greatest detail, this simple opcode is broken down into 27 separately displayed steps. A text window explains what has happened after each step.

PRODUCT NAME
The Visible Computer: 8088

COMPANY
Software Masters, Inc.

ADDRESS
3330 Hillcroft, Suite BB
Houston, TX 77057

TELEPHONE
713/266-5771

PRICE
$69.95

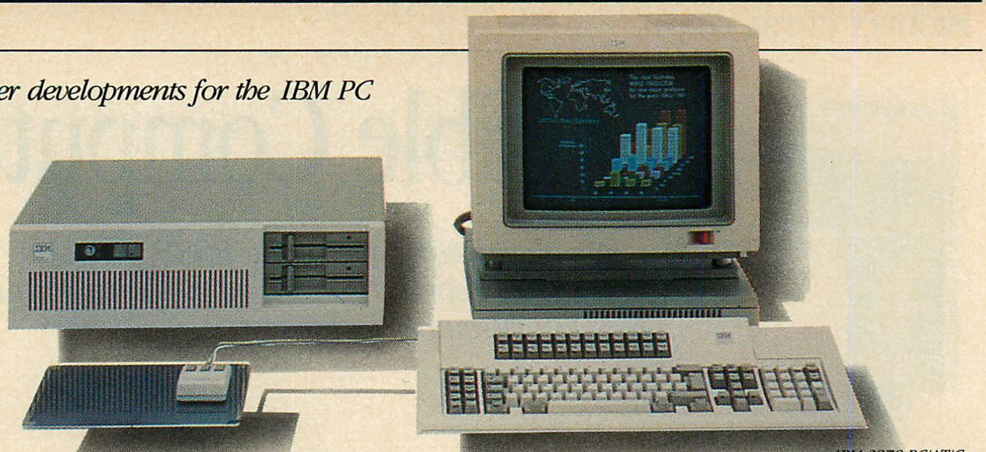
Included on the disk are several programs in 8088 machine code, provided as companions to the tutorial text; they are ready to be loaded and debugged or simulated. However, TVC will operate on any .COM or .EXE file. (TVC failed the acid test: running itself under its own simulator. It may be that the product can correctly simulate any program that does not attempt to set its own breakpoints, nor insist upon operating in realtime.)

For the difficulty of the material, the text is written in a crisp, informal style that bespeaks a knowledge of and a genuine interest in the subject. (The author is not identified.) The 340-page manual is neat, though not typeset, and includes a reprint of Intel's 8088 instruction set reference. Its frequent diagrams are well drawn and helpful. The subject matter is tied fairly tightly to the IBM PC hardware and TVC works with any version of PC-DOS, including 1.1. Its only deficiency is that it was not designed to understand path names.

Obviously more than just a tutorial for 8088 assembly language, TVC teaches the architecture of the 8088. The user "sees" the 8088 as the machine that it is. Assembly language is conceptually similar among CPUs; any version generally includes moves, shifts, logical operations, arithmetic operations, and other odds and ends. Most seasoned assembly language programmers can learn a new assembly language without much difficulty. CPU architectures, however, are less standard and more difficult to understand; yet, intimate knowledge of CPU architecture can make the (important) difference between competence and excellence.

The Visible Computer: 8088 is a good tool because it is at once useful and educational, offering an interactive debugger and a self-paced course in 8088 architecture and assembly language. It treats the 8088 as a machine, not simply a mnemonic device. The software is not copy-protected, and for the $69.95 price tag, this package is a considerable value. Even the competent 8088 assembly language programmer should not miss it.

Hardware, software, and other developments for the IBM PC



IBM 3270-PC/AT/G

HARDWARE

IBM has released the **3270 Personal Computer AT** which doubles the processing power of the 3270-PC. It is available with 512KB and 640KB of memory and 1.2MB of disk storage. Fixed-disk storage is expandable to 40MB. The 3270 PC/AT permits users to display data from four host computers, multiple PC sessions, and two notepads. $7,385.

Two graphics versions of the 3270 PC/AT—the **3270-PC/AT/G** and **AT/GX**—are also available. They can access data from System 370 host computers to create and display color graphics images as well as text. 3270-PC/AT/G, $11,670; 3270-PC/AT/GX, $18,920.

IBM System/36-PC

Also announced was the **IBM System/36 PC** which combines a new entry-level System/36 processor (the 5364) and a directly attached IBM PC. The new System/36 PC allows several users to simultaneously share data and programs from both the System/36 and the PC environments. Four local workstations can be supported by the System/36 PC, including the directly attached PC, system printer, and any combination of two PCs, low-cost PC printers, or System/36 displays and printers. As many as 64 remote workstations can be attached. $5,995 (256KB of memory and 40MB of fixed-disk storage). IBM has also reported price reductions of between 9 and 23 percent on its **System/36 5360** and **5362** processors.

Finally, two new ASCII terminals have been announced. The **IBM 3161** is a high-function, entry-level ASCII display with built-in capabilities to emulate five non-IBM ASCII terminals: ADDS Viewpoint, Hazeltine 1500, Lear Siegler ADM3A and ADM5, and Televideo 910, as well as the IBM 3101. The **IBM 3163** also emulates the 3101 and uses a cartridge that is plugged in to emulate DEC VT52 and VT100 terminals. $695 each.
IBM, Information Systems Group, 900 King Street, Rye Brook, NY 10573; 914/934-4834
CIRCLE 304 ON READER SERVICE CARD

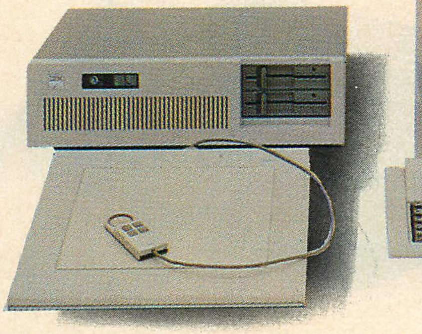
The **Turbo II** from **ALF Products, Inc.** is a diskette duplicator for medium-to-large-volume production of 5¼-inch disks. It copies IBM PC formats, as well as virtually all other soft-sector 5¼-inch formats. $1,450 for basic system; $5,600 for maximum system.
ALF Products, Inc., 1315F Nelson Street, Denver, CO 80215; 303/234-0871
CIRCLE 305 ON READER SERVICE CARD

Fourteen **color** and **monochrome monitors** have been introduced by **Thomson Consumer Products Corporation**. Eight of the monitors are professional models that offer nonglare, tinted screens, built-in speakers, a text display area of 25 lines by 80 characters, and both composite and RGB inputs. Five color monitors and one high-resolution monochrome monitor are also offered. $139 to $449.
Thomson Consumer Products Corporation, 330 Washington Street, Suite 509, Marina del Rey, CA 90292; 213/821-2995
CIRCLE 306 ON READER SERVICE CARD

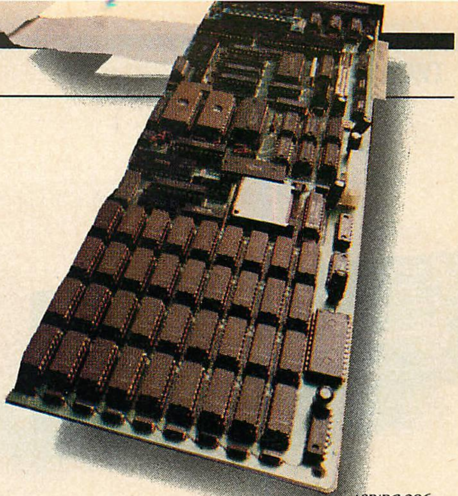
Kustom Electronics, Inc. has introduced an addition to its line of removable hard-disk systems, the **Sunflower MS 10/10 RO.** The MS 10/10 is a dual 10MB unit designed specifically for the handling of sensitive or classified data. This removable cartridge system allows multiple users to employ a hard disk and maintain their individual databases for the cost of a hard-disk cartridge rather than that of a hard-disk drive. The MS 10/10 is transparent to DOS; no additional software is required. $3,395.
Kustom Electronics, Inc., 8320 Nieman Road, Lenexa, KS 66214-1510; 800/445-DISK
CIRCLE 353 ON READER SERVICE CARD

DecaTek, Inc. has introduced a fully secure, dial-up, 9600 synchronous modem called **ZIPmodem**. It is both a high-speed modem and an SNA/SDLC protocol converter on a single IBM plug-in board. ZIPmodem has an on-board processor that frees the PC to run programs while maintaining host connection; it conforms to CCITT V.29 and V.27 specifications and offers error monitoring with automatic fallback to lower speeds if necessary. A stand-alone mainframe version with an NRZI switch option is available. Internal PC version, $2,995; stand-alone version, $1,995.
DecaTek, Inc., P.O. Box 569, Stone Mountain, GA 30086-0569; 404/493-7273
CIRCLE 308 ON READER SERVICE CARD

A new 1200-baud, Hayes-compatible modem called the **FasTalk 1200** has been introduced by **Universal Data Systems, Inc.** Intended for VADs and VARs, this full-duplex modem comes in both a desktop and a plug-in version. In addition to full Hayes compatibility, the FasTalk 1200 offers features that are not available on Hayes products. For example, a voice (or other nondata) signal received during a data call is automatically routed through the modem's speaker and suspends data transfers automatically. $525.
Universal Data Systems, Inc., 5000 Bradford Drive, Huntsville, AL 35805-1953; 205/837-8100
CIRCLE 309 ON READER SERVICE CARD



IBM 3270-PC/AT/GX



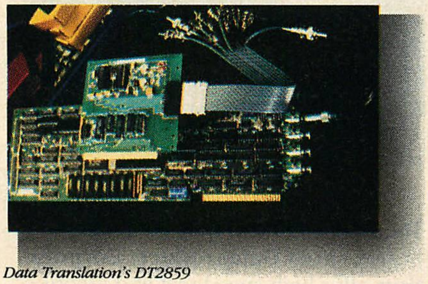
ASP/PC-286

Sunol Systems has introduced the availability of 110MB of storage for its SUN•DISK Winchester 5¼-inch hard-disk drive. The SUN•DISK drive uses Sunol's proprietary run-length-limited encoding technique, which increases usable storage capacity as much as 60 percent over other methods. A random-access tape back-up system is also available. 110 MB with tape backup, $8,354; without tape backup, $6,995.
Sunol Systems, P.O. Box 1777, 1187 Quarry Lane, Pleasanton, CA 94566; 415/484-3322
CIRCLE 311 ON READER SERVICE CARD

The POLYGONE graphics expansion board from **Xtar Electronics, Inc.** features a polygon-filling speed of 100 million pixels per second and is intended for OEM applications, such as CAD, animation, and simulation. The POLYGONE board plugs directly into a single slot of a PC, PC/XT, or PC/AT; it includes an on-board, 16KB-by-16-bit RAM display list, expandable to 64KB by 16 bits, and a 640-by-400-by-4 double frame buffer. The board can display 16 solid colors and 128 dithered patterns simultaneously, out of a palette of 4,096 colors. Its VSR circuits provide frame buffer read-write-modify operations. $3,500.
Xtar Electronics, Inc., 2262 Landmeier Road, Elk Grove, IL 60007; 312/364-4111
CIRCLE 312 ON READER SERVICE CARD

A new upgrade kit from **Dresselhaus Computer Products** adds a near-letter-quality capability to Epson FX printers. The **Finger Print "LetterWriter" NLQ** upgrade kit provides four times the resolution of standard printing and makes the FX printer compatible with software that uses the IBM Graphics printer's special character set. $79.95.
Dresselhaus Computer Products, 837 E. Alosta Avenue, Glendora, CA 91740; 818/914-5831
CIRCLE 313 ON READER SERVICE CARD

Data Translation, Inc. has introduced an eight-channel video multiplexer for the PC. The **DT2859** is a half-size expansion card that multiplexes as many as eight RS-170, NTSC, RS-330, CCIRR (50Hz), or PAL (50Hz) video camera signals for input to the company's DT2803 Frame Grabber, a frame grabber for realtime digital image processing that features 6-bit digitization per pixel, look-up tables, on-board frame storage, and RGB output. The DT2859 also provides two identical video outputs: one output may drive a slave monitor for previewing scenes in undigitized video, while the second output drives a DT2803 Frame Grabber for processing previously acquired images. $359.
Data Translation, Inc., 100 Locke Drive, Marlboro, MA 01752; 617/481-3700
CIRCLE 314 ON READER SERVICE CARD



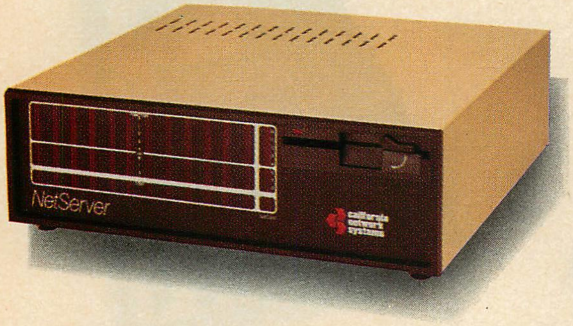
Data Translation's DT2859

The **ASP/PC-286** module has been announced by **American System Products, Inc.** Available in 6-, 8-, and 10-mHz versions, the 286-based ASP modules provide PC or PC/XT users with multitasking, multiuser capabilities. The ASP/PC-286 can provide four RS-232 ports and a full megabyte of memory. The 8088 processor on the PC's main board is replaced by an expansion plug, to which the ASP/PC-286 board is connected; the ASP/PC-286 board requires a full slot. $1,297.
American System Products, P.O. Box 1620, Casselberry, FL 32707; 305/774-4070
CIRCLE 307 ON READER SERVICE CARD

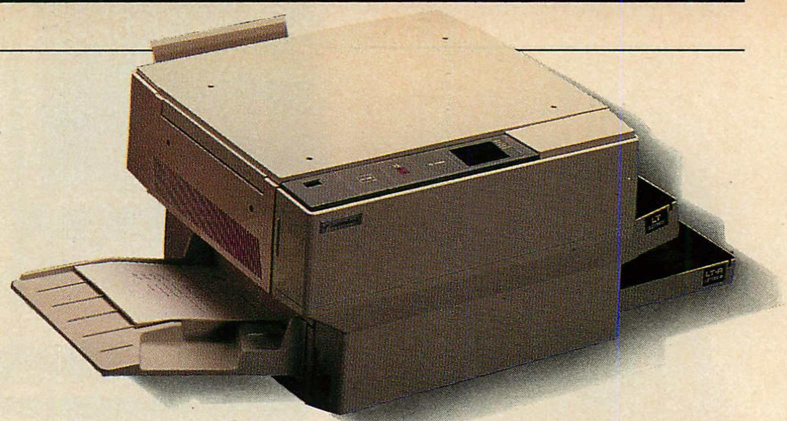
The GAMMA-NET **3.1** from **Gamma Productions, Inc.** connects a PC, PC/XT, or PC/AT with MS-DOS S-100 systems. The 2MB-per-second controllers boast 300 to 400KB-per-second data throughput and can link as many as 255 nodes using bus topology and CSMA/CA/CD access methods. An optional boot eprom allows PCs equipped with GAMMA-NET to become diskless workstations. $299.95, including background disk-server software.
Gamma Productions, Inc., 817 10th Street, Suite 102, Santa Monica, CA 90403; 213/451-9507
CIRCLE 315 ON READER SERVICE CARD

Ariel Corporation has introduced the **PC FFT**, a low-cost dedicated array processor that performs fast Fourier transforms (FFTs) with great speed. The board plugs into a single expansion slot in the PC, PC/XT, or PC/AT, and it processes FFTs 100 to 10,000 times faster than FFT-processing software does and 5 to 30 times faster than general-purpose array processors or 320-type processors do. PC FFT supports several languages: interpreted and compiled BASIC, IBM Pascal and FORTRAN, Lattice C, and Turbo Pascal. $1,850.
Ariel Corporation, 600 W. 116th Street, Suite 84, New York, NY 10027; 212/662-7324
CIRCLE 316 ON READER SERVICE CARD

AST Research, Inc. has announced the **AST FourPort/XN** adapter, a single-slot board that features XENIX device drivers and diagnostics, allowing a PC/AT running XENIX to support four devices or users. The XENIX device-driver software supplied with the adapter can support two four-port boards simultaneously, thus overcoming the AT's hardware limit of three ports or users. The diagnostics ensure that peripherals are properly configured. $495.
AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714; 714/863-1333
CIRCLE 318 ON READER SERVICE CARD



NetServer

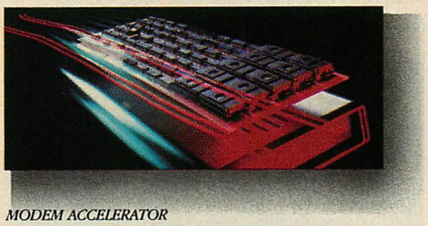


LZR 2650

The first open-architecture LAN-to-mainframe connectivity system predicated upon the IBM NETBIOS session-layer interface has been introduced by **California Network Systems.** The **PCNetwork/SNA 3270 GateWay** works with the IBM PC Network. It allows each PC on the network to operate as a workstation emulating IBM 3270 display stations and printers while retaining the ability to concurrently run other DOS applications. Either a PC or a CNS NetServer Communications System may be used as the PCNetwork/SNA 3270 GateWay, providing a connection to the IBM mainframe as if it were an IBM 3274 Cluster Controller connected via an SNA/SDLC link. Costs for the system run to less than $500 per workstation.
California Network Systems, 372 Turquoise Street, Milpitas, CA 95035; 408/943-9280
CIRCLE 301 ON READER SERVICE CARD

A new circuit board that interfaces with video recording machines has been introduced by **Kirsch Technologies, Inc.** The **Video-Memory Manager** plugs into any expansion slot in the PC or PC/XT and operates with standard video signals, allowing any VCR (beta or VHS format) or optical disk to be used. The board converts digital data into an analog signal, which allows the operator to interact easily with video pictures and software-generated prompts. A unique error-detection scheme allows up to 96MB of data to be stored reliably on one video cassette tape. A utility program called VIDEO-DATA FILER is provided to allow the user to back up and restore memory among floppy disks, hard-disk systems, and video tape. $650; with automatic tape positioning, $875. Both products include Background Back-up software.
Kirsch Technologies, Inc., 201 N. Riverside Drive, St. Clair, MI 48079; 313/329-7166
CIRCLE 317 ON READER SERVICE CARD

The **MODEM ACCELERATOR Multifunction Card** from **Datran** provides telecommunications that are three to four times faster than normal when it is used in conjunction with virtually any modem and software capable of sending binary files. English-language text and other ASCII data are compressed to between one-fourth and one-third their original size; they can then be stored in less space and transmitted in less time. Because this compression takes place in hardware, 500 words can be processed per second, and realtime communications throughput rates can be in excess of 24000 baud. $795.
Datran, 10519 Lauriston Avenue, Los Angeles, CA 90064; 213/474-3684
CIRCLE 321 ON READER SERVICE CARD



MODEM ACCELERATOR

The **LZR 2650** and **LZR 2655** laser printers from **Dataproducts Corporation** offer high-speed text and graphics printing. Intended to serve the needs of computer-aided publishing and typesetting, CAD/CAM, and art and business graphics applications, the LZR 2650 and 2655 print 26 pages per minute at a high resolution of 300 dots per inch. The LZR 2655 supports a single, ledger-sized image (11 by 17 inches) or dual-buffered letter-sized images, and the LZR 2650 supports single letter- and legal-sized images. Options include digitized fonts and customized interfaces. LZR 2650, $11,080; LZR 2655, $12,320 (in OEM quantities of 100).
Dataproducts Corporation, 6200 Canoga Avenue, Woodland Hills, CA 91365; 213/887-8451
CIRCLE 310 ON READER SERVICE CARD

Optimem's Optimem 1000 is a laser-based optical disk drive that stores 1 billion bytes of data on one side of a single 12-inch removable disk using write-once technology. It is designed to be a low-cost, mass-storage device for advanced, image-oriented office systems, transaction processing, data archiving, and other data-storage applications. $13,600 (OEM price); 12-inch disk, $400.
Optimem, 435 Oakmead Parkway, Sunnyvale, CA 94086; 408/737-7373
CIRCLE 322 ON READER SERVICE CARD

Number Nine Computer Corporation has introduced three low-power, plug-in graphics boards for the PC/AT. **Revolution 512 x 32** features 1MB of multiported display memory configured to an addressable 512-by-512-pixel resolution, 32 bit planes deep. A quarter million colors may be viewed simultaneously from a palette of more than 16 million. The controller allows software to tell the host processor to access the display buffer directly for fast drawing or other capabilities. $3,995.

Revolution 1280 x 4 features 1MB of multiported display memory configured to an addressable window of 2,048-by-2,048-pixel resolution and a viewable, noninterlaced window of 1,280 by 1,024 pixels, 4 bit planes deep. The controller is the first to offer 100-mHz bandwidth which permits flicker-free display and high resolution. Prices by viewable resolution: 640-by-480, 832-by-624, $1,995 each; 1,024-by-768, $2,995; 1,280-by-1,024, $3,995.

Finally, **Revolution 1024 x 8** offers 1MB of multiported display memory configured to an addressable 1024-by-1024-pixel resolution, 8 bit planes deep. The board allows 256 colors to be viewed simultaneously, selected from a palette of more than 16 million. $3,995.
Number Nine Computer Corporation, 691 Concord Avenue, Cambridge, MA 02138; 617/492-0999
CIRCLE 320 ON READER SERVICE CARD

Available for IBM PC

What C did for Programming Mark Williams has done for C Programming

The C Programming System from Mark Williams

MWC86 gets your C programs running faster and uses less memory space than any other compiler on the market. Then *csd*, Mark Williams' revolutionary C Source Debugger, helps you debug faster. That's The C Programming System from Mark Williams Company.

MWC86

MWC86 is the most highly optimized C compiler available anywhere for the DOS and 8086 environment. The benchmarks prove it! They show MWC86 is unmatched in speed and code density.

MWC86 supports large and small models of compilation, the 8087 math coprocessor and DOS 2.0 pathnames. The compiler features common code elimination, peephole optimization and register variables. It includes the most complete libraries. Unlike its competition, MWC86 supports the full C language including recent extensions such as the Berkeley structure rules, voids, enumerated data types, UNIX* I/O calls and structure assignments.

Quality is why Intel, DEC and Wang chose to distribute MWC86. These industry leaders looked and compared and found Mark Williams to be best.

User Friendly

MWC86 is the easiest to use of all compilers. One command runs all phases from pre-processor to assembler and linker. MWC86 eliminates the need to search for error messages in the back of a manual. All error messages appear on the screen in English.

A recent review of MWC86 in *PC World*, June, 1984, summed it up:

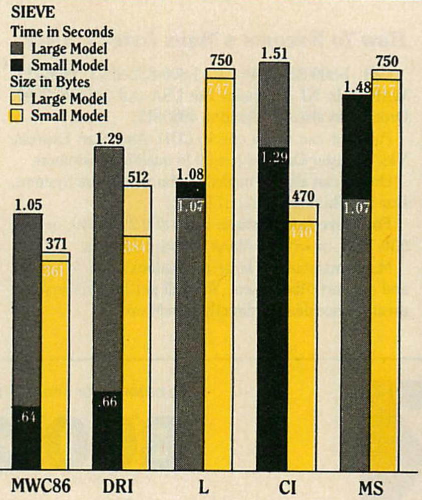
"Of all the compilers reviewed, MWC86 would be my first choice for product development. It compiles quickly, produces superior error messages, and generates quick, compact object code. The library is small and fast and closely follows the industry standard for C libraries."

csd C Source Debugger

Mark Williams was not content to write the best C compiler on the market. To advance the state of the art in software development, Mark Williams wrote *csd*.

csd C Source Debugger serves as a microscope on the program. Any C expression can be entered and evaluated. With *csd* a programmer can set tracepoints on variables and expressions with full history capability and can single step a program to find bugs. The debugger does not affect either code size or execution time. *csd* features online help instructions; the ability to walk through the stack; the debugging of graphics programs without disturbing the program under test; and evaluation, source, program and history windows.

csd eases the most difficult part of development — debugging. Because *csd* debugs in C, not assembler, a programmer no longer has to rely on old-fashioned assembler tools, but can work as if using a C interpreter — in real time.



The C Programming System from Mark Williams now supports the following third party software:

Windows for C, Halo, PHACT, The Greenleaf Functions, Btrieve, Panel, Plink 86, cView, C Utility Library, C Tree, The Application Programmer's Tool Kit, C Power Packs, PCLOCATE, Super C Index System, Opt-Tech Sort, Safe C Interpreter, C-terp, ProScreen, BRIEF, C Helper.

The C Programming System from Mark Williams

The C Programming System from Mark Williams delivers not only the best C compiler for the 8086 but also the revolutionary *csd* C source level debugger. That's why it does for C programming what C did for programming. The Mark Williams C Programming System gives the programmer the MWC86 C compiler and the *csd* C Source Debugger for only $495. Order today by calling 1-800-MWC-1700. Major credit cards accepted.

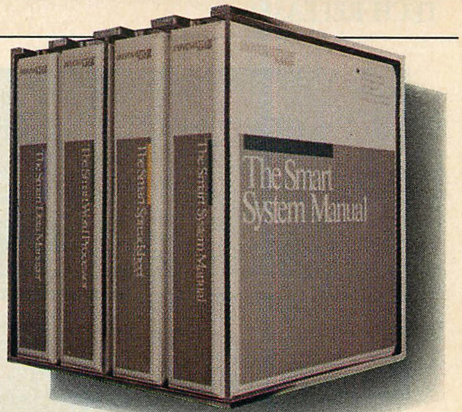
Technical support for The Mark Williams C Programming System is provided free of charge by the team that developed it.

Mark Williams Company
1430 W. Wrightwood Ave.
Chicago, IL 60614

CIRCLE NO. 186 ON READER SERVICE CARD

*Unix is a Trademark of Bell Laboratories.

Reel-Core 7



The Smart Software System 2.0

Lancore Technologies, Inc.'s **Reel-Core 7** is a 9-track, reel-to-reel magnetic tape storage system that can read or write ½-inch, IBM-compatible 1,600- or 3,200-bpi phase-encoded tape. Reel-Core 7 not only operates as a back-up system for a Winchester hard disk but it also can write mainframe-compatible tape reels for file transport from PC to mainframe and vice versa. $4,495.
Lancore Technologies, Inc., 31324 Via Colinas, Westlake Village, CA 91361; 818/991-5100
CIRCLE 325 ON READER SERVICE CARD

A similar tape storage system is offered by **Emerald Systems Corporation**. The **PS46-7000A** (automatic loading) and the **PS-7000M** (manual loading) are two versions of a ½-inch tape subsystem that converts data from EBCDIC to ASCII format and into the formats used by popular spreadsheet, database, and other PC programs. Micro users can download mainframe datasets off-line into PC formats, such as those for Lotus 1-2-3, dBASE III, or Symphony. The system selectively sorts records prior to the initial conversion, so that users can download only data that are important to their applications. PS46-7000A, $9,250; PS-7000M, $5,950.
Emerald Systems Corporation, 4757 Morena Blvd., San Diego, CA 92117; 619/270-1994
CIRCLE 326 ON READER SERVICE CARD

Reference Technology, Inc. has announced the **CLASIX DataDrive Series 500**, a desktop read-only laser optic peripheral that delivers large databases (550 million characters) on prerecorded media to PC users. The new system is based on CD ROM technology and supports the CLASIX CD, a 12-centimeter prerecorded optical disk. $1,535.

The company also announced that its **TRIDECC** data-preparation services have been extended to support the production of CLASIX CD disks. Data can be rapidly converted to either the 2-billion-byte, 12-inch CLASIX DataPlate or the 550-million-byte, 12-centimeter CLASIX CD. $8,000 per disk side, plus $250 per additional reel of input data tape.

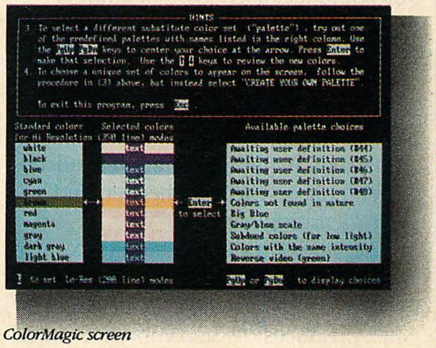
In addition, the company's **STA/F File** software package has been enhanced to provide full support for the CLASIX CD. STA/F File optimizes and standardizes the access to and format of large volumes of data published on read-only optical disks, thereby allowing databases to be used on various systems, independent of operating environments. The software also extends the addressing range of PC-DOS to enable standard PC applications to access databases as large as 4 billion bytes. $110; CLASIX CD replicas, $15 per copy.
Reference Technology, Inc., 1832 N. 55th Street, Boulder, Co 80301; 303/449-4157
CIRCLE 323 ON READER SERVICE CARD

The **color hard copier** from **RACET computes, Ltd.** provides very high resolution, high color density copies on plain paper or directly on transparency materials. The RACET color hard copier can be connected directly to the RGB output, eliminating the need for any special software drivers.

Also from RACET comes the **PCMS-240**, a 240MB mass-storage subsystem for the PC, PC/XT, and PC/AT. The PCMS-240 provides storage that is transparent to MS-DOS and allows large central files (greater than 32MB) to be created with password and read/write protection. Other features include dynamic partitioning (up to 800 user partitions), back-up and restore utilities, and remote diagnostic utilities. The PCMS-240 supports 3-COM Ethernet, Novell, IBM PC Net, and RACET MUXNET. Prices: RACET color hard copier, $5,495; PCMS-240, $19,500.
RACET computes, Ltd., 1855 W. Katella, Orange, CA 92667; 714/997-4950
CIRCLE 319 ON READER SERVICE CARD

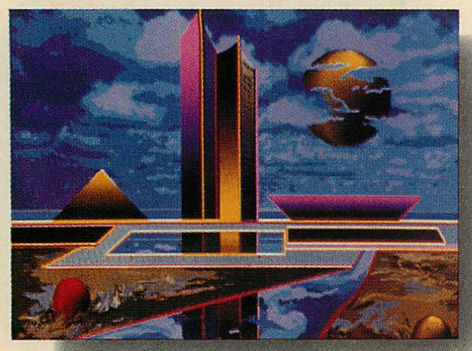
SOFTWARE

ColorMagic, from **Lifetree Software, Inc.**, is a color-enhancement software package that works in conjunction with the IBM Enhanced Graphics Adapter and any color monitor to increase the number of colors accessible from 16 preselected colors to any 16 of 64 available colors. Users may choose from 15 preconfigured color palettes or they may create as many as 48 palettes of their own. $39.95.
Lifetree Software, Inc., 411 Pacific Street, Monterey, CA 93940; 408/373-4718
CIRCLE 327 ON READER SERVICE CARD

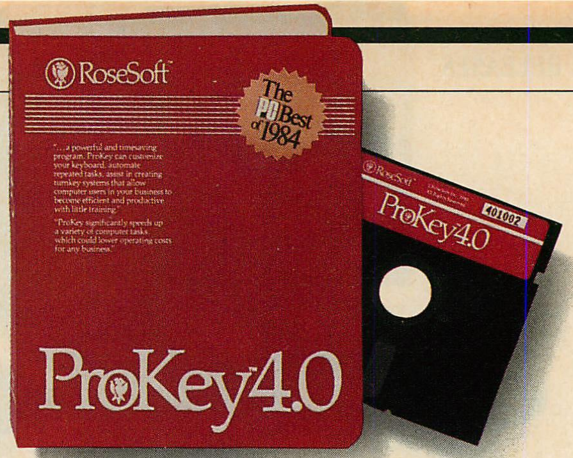


ColorMagic screen

Innovative Software has introduced **The Smart Software System 2.0**, a modular, integrated system that includes a communications module, a new programming language that is more powerful than dBASE III, spreadsheet enhancements that offer users greater capacity and speed than Lotus 1-2-3, a word processor, and a time manager. Word-processing module, $295; Smart Spreadsheet, $395; database, $495; fully integrated system, $895. Upgrades for owners of Multimate, WordStar, Lotus 1-2-3, dBASE, and Symphony: $195 per program; $495 for complete system.
Innovative Software, 9300 W. 110th Street, Suite 380, Overland Park, KS 66210; 913/345-2424
CIRCLE 335 ON READER SERVICE CARD



Brushwork screen



RoseSoft's ProKey 4.0

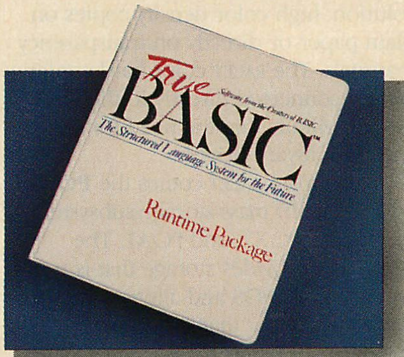
QUEPRO, from **Bytel Corporation**, is a menu-driven, interactive report generator that allows managers to design their own report layouts on the screen. Once a programmer customizes QUEPRO by defining the relationship between the company's data files and the data elements they contain, a manager may use the program's runtime processor to access data quickly. $595.
Bytel Corporation, 1730 Solano Avenue, Berkeley, CA 94707; 415/527-1157
CIRCLE 328 ON READER SERVICE CARD

A product that provides programming capability for users of Lotus 1-2-3 has been announced by **Fox & Geller**. **QUICKMAC** lets users type in familiar programming statements, such as DO . . . WHILE and IF . . . THEN . . . ELSE, while they are using 1-2-3. From these statements, QUICKMAC generates complete programs using 1-2-3's built-in macro facility. Included are a full-featured screen editor, a syntax-checker, and a macro learning facility. $79.

Another product, **QUICKINDEX**, provides a high-speed indexing capability for users of dBASE II and dBASE III. QUICKINDEX produces dBASE-compatible index files up to 10 times faster than dBASE. $69 for copy-protected version; $99 for non-copy-protected version.
Fox & Geller, Inc., 604 Market Street, Elmwood Park, NJ 07407; 201/794-8883
CIRCLE 329 ON READER SERVICE CARD

Microtec Research, Inc. has announced **Professional Pascal**, a resident and cross compiler that runs on all processors in the 8086 family and can generate optimized code for any of the other processors in that family, including the special instructions required by the 80186 and 80286. A DEC VAX-resident compiler is also available; it allows programmers to develop and test software on a DEC VAX and then cross-compile it for downloading to an 8086 family processor. Professional Pascal supports error-reporting and recovery and offers an extensive set of runtime library routines and utilities. $895 for MS-DOS license; $5,000 for DEC VAX/VMS or DEC VAX/UNIX license.
Microtec Research, Inc., 3930 Freedom Circle, Suite 101, Santa Clara, CA 95054; 408/733-2919
CIRCLE 330 ON READER SERVICE CARD

A companion product to the True BASIC Language System—the **Runtime Package**—has been introduced by **True BASIC, Inc.** The Runtime Package binds compiled True BASIC programs with a runtime module to produce compact, stand-alone code that uses less memory and loads faster than code created with the True BASIC Language System. A library of routines that provides access to the PC's BIOS and DOS routines is also included. $500.
True BASIC, Inc., 39 S. Main Street, Hanover, NH 03755; 603/643-3882
CIRCLE 337 ON READER SERVICE CARD



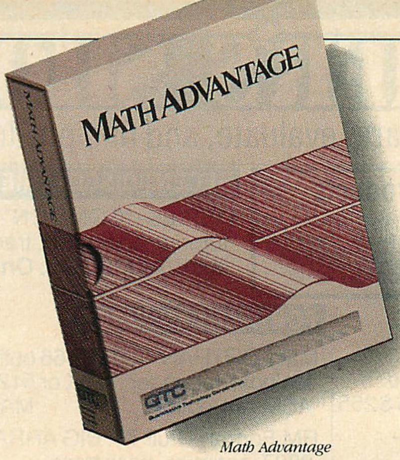
Runtime Package by True BASIC

LANLink, from **The Software Link, Inc.**, is a software-driven LAN for the PC. Network interface boards need not be installed in PCs to set up a LANLink network; instead, RS-232 ports are used for all network communications and LANLink's server and satellite disks contain the logic normally found on network boards. LANLink features the ability to link multiple servers for hard-disk backup and to interact with the network via modem in realtime. $495.
The Software Link, Inc., 8601 Dunwoody Place, Suite 336, Atlanta, GA 30338; 404/998-0700
CIRCLE 332 ON READER SERVICE CARD

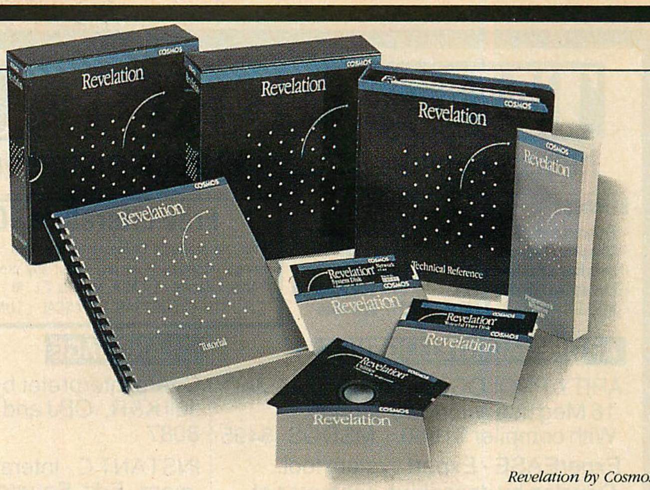
Version 4.0 of **RoseSoft, Inc.**'s **ProKey** macro processor has been announced. This version enables users to name macros with meaningful words rather than single characters or two-key combinations. In addition, the package provides the capability to create guarded macros that cannot be overwritten. $129.95.
RoseSoft, Inc., 4710 University Way NE, Suite 601, Seattle, WA 98105; 206/524-2350
CIRCLE 331 ON READER SERVICE CARD

West End Film, Inc. has announced a pixel-based interactive painting package with an unlimited number of brushes, extensive shading and tinting capabilities, and the ability to scan in images using a video camera. **Brushwork** supports plotters, ink-jet printers, tablets, mice, and color cards from Number Nine, Scion, Tecmar, and IBM. $1,500.
West End Film, Inc., 2121 Newport Place NW, Washington, DC 20037; 202/331-8078
CIRCLE 333 ON READER SERVICE CARD

Invisible Optimizer, a disk-buffering product for enhancing the speed of the PC, PC/XT, and PC/AT, has been announced by **Stellation Two, Inc.** It uses disk caching to reduce the amount of time spent waiting for either floppy- or hard-disk systems to access information. The user can allocate any amount of unused RAM memory for the caching. $95; $25 for trial version.
Stellation Two, Inc., 26 W. Mission Street, Suite 3, P.O. Box 2342, Santa Barbara, CA 93120; 805/569-3132
CIRCLE 336 ON READER SERVICE CARD



Math Advantage



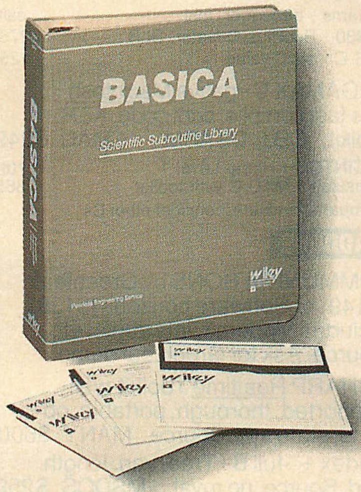
Revelation by Cosmos

Intel 8096 code can be developed on the PC through the use of the new **CYS-8096 Cross Assembler** from **Cybernetic Micro Systems.** The package includes a symbolic assembler, which takes the standard Intel mnemonics as source code input, and generates a listing and Intel-format hex file as output. The Cross Assembler supports program segmentation into code, data, extra, and special function sections, with unique data types assigned to the symbols in each segment. $295.
Cybernetic Micro Systems, P.O. Box 3000, San Gregorio, CA 94074; 415/726-3000
CIRCLE 340 ON READER SERVICE CARD

Peerless Engineering Service has released **PC MAP II**, a new version of the company's topographic mapping and analysis software package for the PC/XT and PC/AT. PC MAP II accepts as many as 5,000 random *XYZ* input points from any ASCII file; it interpolates grids of as many as 65,536 nodes and lets the user specify grids by averaging or "Kriging" (a form of grid interpolation). $995.
Peerless Engineering Service, 5819 Soquel Drive, Soquel, CA 95073; 408/462-0330
CIRCLE 341 ON READER SERVICE CARD

Cosmos, Inc. has introduced a new version of **Revelation**, its database system and applications environment. New features include context-sensitive help screens, more menus, and a file structure that sorts large files up to 10 times faster than previous versions did. $950.
Cosmos, Inc., 19530 Pacific Highway S, Suite 102, Seattle, WA 98188; 206/824-9942
CIRCLE 348 ON READER SERVICE CARD

A new scientific subroutine library called **BASICA** has been introduced by **Wiley Professional Software**. Designed for programmers, scientists, engineers, teachers, and students, the BASICA library solves such problems as matrices with real elements, matrices with complex elements, polynomials, differentiation, numerical integration, and systems of equations. $125.
Wiley Professional Software, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158; 212/850-6788
CIRCLE 342 ON READER SERVICE CARD



BASICA subroutine library

A similar library is offered by **Quantitative Technology Corporation.** The **Math Advantage** is a library of core algorithms that enhances the productivity of engineers and scientists developing numerically intensive applications. Subroutines in the library can be called from FORTRAN or C programs. $495.
Quantitative Technology Corporation, 8700 S.W. Creekside Place, Suite D, Beaverton, OR 97005; 503/626-3081
CIRCLE 343 ON READER SERVICE CARD

Version 2.2 of the **EXSYS Expert System Development Package** has been released by **Exsys, Inc.** The new version allows EXSYS expert systems to call a wide variety of external programs, which can then pass data back into EXSYS for analysis. EXSYS can function as a general inference engine in an integrated approach to artificial intelligence, using other programs for data acquisition and, in some cases, manipulation. $295.
Exsys, Inc., P.O. Box 75158, Contr. Sta. 14, Albuquerque, NM 87194; 505/836-6676
CIRCLE 339 ON READER SERVICE CARD

MATRIX 100, from **Stanford Business Software, Inc.,** is an enhancement of BASIC on the PC. It lets the user perform complex matrix operations with a single BASIC statement for each operation; such operations execute up to 100 times faster than those in BASIC. MATRIX 100 has extended commands for such operations as LU factorizations. $80; $125 with 8087 support.
Stanford Business Software, Inc., 4151 Middlefield Road, Suite 215, Palo Alto, CA 94303; 415/424-9499
CIRCLE 347 ON READER SERVICE CARD

A software product that allows PCs tied to the IBM 3270 network to share mainframe printer resources has been announced by **Avatar Technologies, Inc.** Using **Print-Net**, any number of PCs that are connected by coaxial cable to a 3274 or 3276 control unit can direct printing requests to IBM 3287 printers without additional hardware. $195.
Avatar Technologies, Inc., 99 South Street, Hopkinton, MA 01748; 617/435-6872
CIRCLE 346 ON READER SERVICE CARD

MLINK, a comprehensive system for establishing micro-to-mainframe and micro-to-micro communications, has been released by **Corporate Microsystems, Inc. MLINK** incorporates three file-transfer protocols, built-in on-line help functions, directory-based autodialing for all popular modems, and automatic log-on. The package can perform as a terminal emulator, an error-free data-transfer utility, or a remote bulletin board. $250.
Corporate Microsystems, Inc., P.O. Box 277, Etna, NH 03750; 603/448-5193
CIRCLE 344 ON READER SERVICE CARD

TED MIRECKI

Breakpoints in BIOS

The MOVROM program lets programmers trace through BIOS routines using breakpoints.

The ROM BIOS routines are a treasure trove of information about the nitty-gritty details of IBM PC systems. To understand these routines, it is best to trace through the code with a debugger. The DOS DEBUG program will single-step through the BIOS, but breakpoints cannot be set in ROM because doing so requires that code be replaced by breakpoint instructions. MOVROM.ASM copies the BIOS to RAM, makes it resident, and repoints all BIOS interrupts to the copied code, thus allowing any debugger operations to be performed.

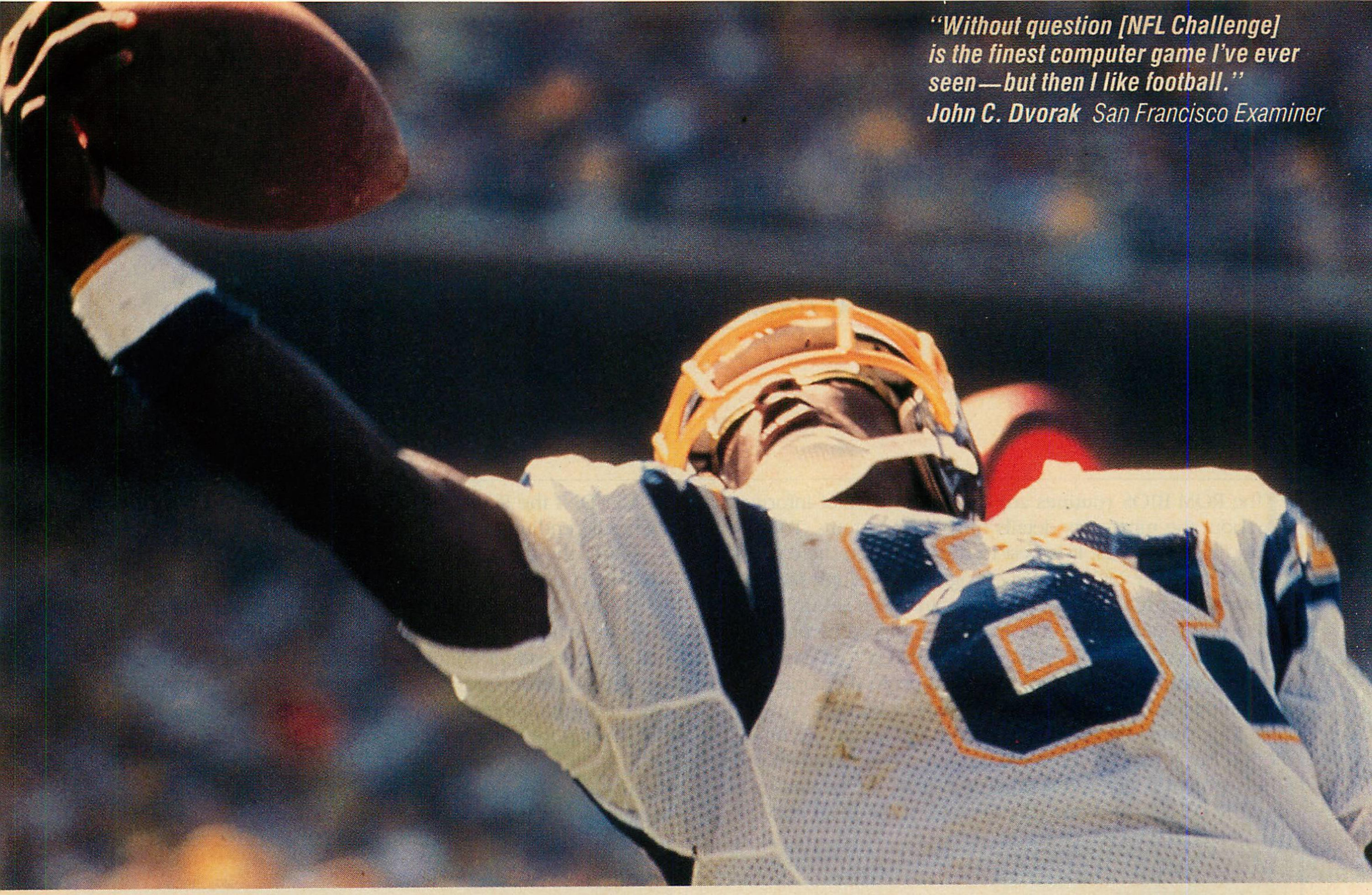
This routine copies the system BIOS for the PC or PC/XT, but it may be modified to copy any other ROM code. The segment is entered at label ROMSEG, the beginning offset at ROMBEG, and one past the end of the ROM code at ROMEND. If the ROM code extends to the end of the segment, the ending offset is entered as zero.

After the program is assembled, it must be linked with the /HIGH parameter. This ensures that the area just above DOS is free to receive the ROM code. The program repoints ROM interrupt vectors to RAM. MOVROM must be executed before any resident utilities are installed, because such utilities repoint BIOS interrupts to themselves. When MOVROM exits, it displays the segment where it copied the ROM code; the offset is the same as in the original BIOS listing.

Ted Mirecki is a corporate planner who is reponsible for developing decision-support systems on a variety of hardware.

LISTING: MOVROM.ASM

```
STACK   SEGMENT STACK
        DW      128 DUP(0)
STACK   ENDS

ASMUTIL SEGMENT BYTE PUBLIC 'CODE'
        ASSUME  CS:ASMUTIL, SS:STACK
ROMSEG  DW      0F000H          ;ROM's segment
ROMBEG  DW      0E000H          ;starting paragraph
ROMEND  DW      0               ;ending paragraph
MESSG   DB      'ROM code moved to segment '
SEG$    DB      0,0,0,0,'$'

MOVROM  PROC    FAR
        MOV     AX,DS           ;get PSP segment
        ADD     AX,6            ;first avail. paragraph
        MOV     BX,CS:ROMBEG    ;get starting offset
        MOV     SI,BX           ;point to source offset
        MOV     DS,CS:ROMSEG    ;get source seg into DS
        MOV     CX,4
        SHR     BX,CL           ;starting paragraph
        SUB     AX,BX
        MOV     ES,AX           ;destination segment
        MOV     DI,SI           ;dest. offset same as source
        MOV     BX,SI           ;save it in BX
        MOV     CX,CS:ROMEND    ;calc word length in CX
        SUB     CX,SI           ;CX is ROM length
        SHR     CX,1            ;convert to word length
        JNZ     SAVLEN
        MOV     CH,80H          ;if zero, set to 1/2 segment
SAVLEN: PUSH    CX              ;save word length
        REP     MOVSW           ;move ROM to low memory
        MOV     AX,DS           ;point AX to ROM segment
        MOV     DX,ES           ;point DX to new segment
        MOV     SI,BX           ;reset SI to start of new code
        XOR     BX,BX           ;DS:BX points to intrpt table
        MOV     DS,BX
        MOV     CX,100H         ;no. of interrupts in table
        DEC     DI              ;DI = last byte of moved code
SCNINT: CMP     AX,[BX+2]       ;int points to ROM segment?
        JNE     NXTINT          ;no, go look at next int
        CMP     [BX],SI         ;int offset past start?
        JB      NXTINT          ;no, go look at next int
        CMP     [BX],DI         ;int offset before end?
        JAE     NXTINT          ;no, next int
        MOV     [BX+2],DX       ;yes, set new segment
NXTINT: ADD     BX,4            ;point to next int
        LOOP    SCNINT
        MOV     AX,CS           ;point ES & DS to code segment
        MOV     DS,AX
        MOV     ES,AX
        ASSUME  DS:ASMUTIL, ES:ASMUTIL
        MOV     AX,DX           ;new segment value to AX
        MOV     DI,OFFSET SEG$  ;point DI to HEX char string
        CALL    AX2HEX          ;convert seg to characters
        MOV     DX,OFFSET MESSG ;display sign-off message
        MOV     AH,9
        INT     21H
        POP     DX              ;restore ROM length in words
        ADD     DX,7            ;round up to next paragraph
        SHR     DX,1            ;convert to paragraphs
        SHR     DX,1
        SHR     DX,1
        ADD     DX,6            ;adjust for prefix
        MOV     AX,3100H        ;fix ROM code in memory
        INT     21H
MOVROM  ENDP
        ASSUME  CS:ASMUTIL
AX2HEX  PROC    NEAR
        CLD                     ;go forward through strings
        MOV     BX,OFFSET HEX   ;point to hex digit chars
        MOV     CX,4            ;will handle 4 nibbles
NXTNIB: ROL     AX,1            ;hi-order nibble to lo-order AX
        ROL     AX,1
        ROL     AX,1
        ROL     AX,1
        MOV     DX,AX           ;save it in DX
        AND     AL,0FH          ;isolate low-order nibble
        XLAT    CS:HEX          ;translate to hex digit
        STOSB                   ;put hex digit into output
        MOV     AX,DX           ;recover AX value
        LOOP    NXTNIB          ;repeat for 4 nibbles
        RET
AX2HEX  ENDP
HEX     DB      '0123456789ABCDEF'
ASMUTIL ENDS
        END     MOVROM
```



Put the NFL at your fingertips.

Hands-on pro football excitement is as close as your keyboard with **NFL Challenge** from Xor. This is the officially licensed NFL action computer game with offensive plays and defensive sets based on the ones in NFL playbooks, plus complete updatable rosters for all 28 NFL teams. All impressively packaged.

Every game can be the Super Bowl when you take the field with Xor **NFL Challenge** software. Feel the pressure as you choose starting lineups, drive downfield against the clock, and go head-to-head, down-by-down against the league's most innovative offenses and formidable defenses. It's the most intense computer simulation of the pro football experience ever devised.

Xor's **NFL Challenge** requires IBM® PC or PC/XT with DOS 2.0, 256K bytes of memory, monochrome monitor with IBM® monochrome card or RGB color monitor with IBM® color graphics adapter. Also runs on the IBM® PC/AT.

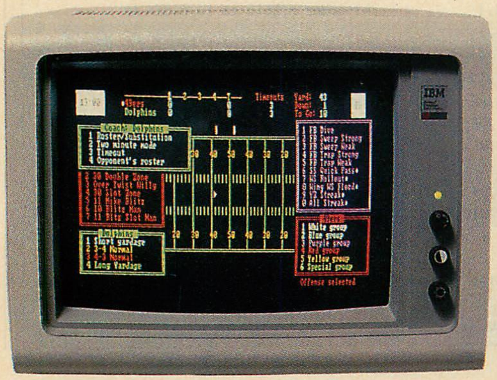
With **NFL Challenge** from Xor, you have 50-yard line seats for every game!

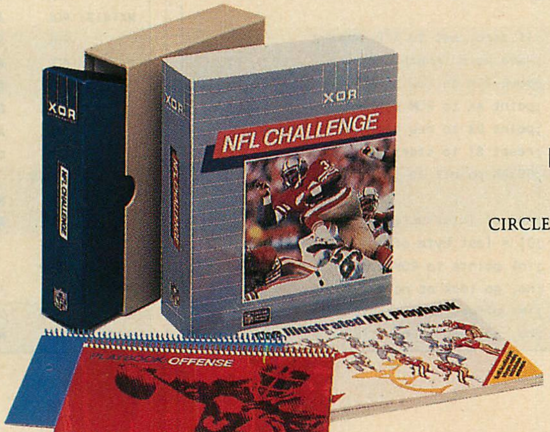
Ordering Information

NFL Challenge will be in short supply. To kick off your NFL season, send us $99.95, or if you're in a hurry-up offense, order by major credit card:

1-800-NFL-CHAL
(in MN call (612) 938-0005)

add $6.00 for shipping to all U.S. cities. Minnesota residents: add 6% sales tax.





XOR™ CORPORATION
5421 Opportunity Court
Minnetonka, Minnesota 55343
(612) 938-0005

CIRCLE NO. 228 ON READER SERVICE CARD

Officially Licensed Product

TECH NOTEBOOK 48

ERNEST S. KOEHLER

Name That Volume

Now DOS 2.x users can create volume labels after their disks have been formatted.

Without volume labels, all disks are very much alike. Unfortunately, unlike DOS 3.0, which has a new LABEL command that allows the user to write a volume label on a disk after it has been formatted, DOS 2.x can write a volume label only during disk format. The program VOLABEL allows DOS 2.x users to create volume labels without reformatting.

Unlike ordinary files, the volume label has a file attribute byte of 08H, and the remainder of the entry contains no useful information. The DOS create file function (16H) creates a file entry in the root directory. The file's extended file control block (FCB) contains an attribute byte that sets the attribute byte in the directory entry. The idea is to set the attribute byte of an extended FCB to indicate a volume label, use the create file function to make a directory entry with the correct attribute, and then close the newly created file so that all that remains is the directory entry, which is now a label.

It is important to note that when DOS function calls that reference an extended FCB are used, the first byte of the extended FCB must be set to FFH, and DS:DX must point to it at offset 55H in the program segment prefix.

The syntax of the VOLABEL command is VOLABEL [d:]. If the drive is specified after the command name, DOS will automatically place the appropriate drive number into the FCB. If the drive is not specified, the default drive is assumed.

However, DOS is not used to enter the volume label from the command line into the FCB. If DOS were used, the volume label would have to be entered using a file-name format, which would be inconvenient for volume labels that have spaces. Instead, the DOS buffered keyboard input function (0AH) is used to put the label into an input buffer; the user then moves the label into the FCB, being careful not to move the carriage return with it.

If a volume label already exists in the root directory, the create file function will not let another one be entered, and the program will respond with an error message.

The VOLABEL program must be assembled and linked. The .EXE file then must be converted to a .COM file using EXE2BIN. VOLABEL should be run as a .COM file.

Ernest S. Koehler is systems analyst for Cornell's school of law.

LISTING: VOLABEL.ASM

```
          PAGE
TITLE     VOLABEL -A VOLUME LABEL WRITING PROGRAM
COMMENT   "         VOLABEL [D:]
          Creates a volume label entry in a root directory.
"
CODE      SEGMENT
          ORG     55H
FLAGBYTE  DB      (?)          ;1ST BYTE OF EXTENDED FCB
          ORG     5BH
ATTRBYTE  DB      (?)          ;ATTRIBUTE BYTE OF EXTENDED FCB SET
DRIVENUM  DB      (?)          ;DRIVE NUMBER
VOLNAME   DB      11 DUP(?) ;VOLUME NAME
          ORG     100H
          ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
VOLABEL   PROC
          ;PRINT PROMPT
          MOV     AH,09H
          MOV     DX,OFFSET PROMPT
          INT     21H
          ;GET VOLUME LABEL NAME
          MOV     AH,0AH
          MOV     DX,OFFSET INBUFF
          INT     21H
          ;MOVE VOLUME NAME TO EXTENDED FCB
          XOR     BX,BX          ;CLEAR BX
          MOV     CX,0BH         ;MAXIMUM LOOP ITERATIONS IS 11
NAMELOOP:
          MOV     AH,[INBUFF+BX+2]    ;MOVE CHAR FROM INPUT BUFF
          CMP     AH,0DH         ;EXIT LOOP IF CHARACTER IS A
                                 ;CARRIAGE RETURN
          JE      LOOPDONE
          MOV     [VOLNAME+BX],AH       ;PLACE CHARACTER IN FCB
          INC     BX             ;POINT TO NEXT CHARACTER
          LOOP    NAMELOOP
LOOPDONE:
          MOV     FLAGBYTE,0FFH    ;SET EXTENDED FCB FLAG
          MOV     ATTRBYTE,08H     ;SET ATTRIBUTE BYTE OF
                                   ;EXT FCB FOR VOLUME LABEL
          ;CREATE DIRECTORY ENTRY FOR VOLUME NAME
          MOV     AH,16H
          MOV     DX,55H      ;SET DX TO START OF EXTENDED FCB
          INT     21H
          ;TEST FOR CREATION ERROR
          OR      AL,AL       ;JUMP IF DIRECTORY ENTRY IS CREATED
          JZ      CREATE_OK
          MOV     AH,9H       ;PRINT CREATION ERROR MESSAGE
          MOV     DX,OFFSET CREATEMSG
          INT     21H
          JMP     VOLABEL_END ;EXIT PROGRAM
CREATE_OK:
          ;CLOSE FILE CREATED BY DIRECTORY ENTRY CREATION
          MOV     AH,10H
          MOV     DX,55H        ;SET DX TO START OF EXTENDED FCB
          INT     21H
VOLABEL_END:
          INT     20H           ;RETURN TO DOS
VOLABEL   ENDP
INBUFF    DB      0CH           ;INPUT BUFFER FOR LABEL NAME
          DB      12 DUP (0)
PROMPT    DB      0DH,0AH,
          'PLEASE ENTER VOLUME LABEL NAME (11 CHARS. MAX): $'
CREATEMSG DB      0DH,0AH,
          'UNABLE TO ENTER VOLUME LABEL INTO DIRECTORY$'
CODE      ENDS
          END     VOLABEL
```

JOSEPH G. LOUDERBACK

Rapid Screens in Turbo

Turbo Pascal can generate instantaneous screen output without eating up space in the data segment.

Turbo Pascal offers several ways to do very fast screen output. You can store screen images in arrays of records and move them into and out of screen buffers, but arrays take up space in Turbo's 64KB data segment.

The listing below, RAPIDSCR.PAS, offers a fast alternative that does not deplete the data segment. The technique stores each screen image in the heap. The size of the heap is limited (roughly) only by the physical memory of the PC. As soon as the images are in the heap, it is a simple matter to recall any of them to the screen by using the MOVE command, which will fill the screen instantly.

RAPIDSCR first writes four screens using Write statements, then saves each image in the heap. Once all screens have been saved, pressing any key loads them to the screen from heap immediately.

For brevity's sake, RAPIDSCR generates its screens from string literals embedded in the program code. The wiser course is to store text screens as text files on disk, and then read them from disk and immediately store them out to the heap. This method does not even require space in the code or data segment for string literals, as demonstrated here. It also allows the user to change screen displays without recompiling the program.

In a modern PC that contains 640KB RAM, the largest memory resource in a Turbo Pascal program is the heap. Used wisely, the 64KB limitation of the data segment becomes much less important.

Joseph G. Louderback is a professor of accounting at Clemson University. He has written several books about accounting and finance.

LISTING: RAPIDSCR.PAS

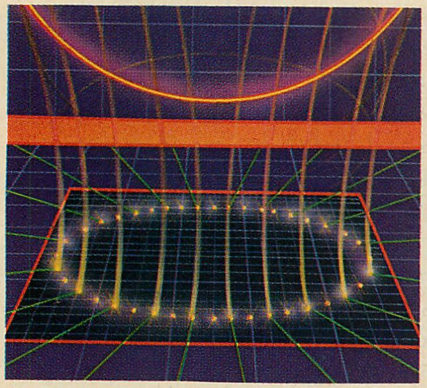
```
{$C-,U-}
program RapidScreen;

TYPE StuffInHeap = ^PutInHeap;           {Pointer to screens}
     PutInHeap = Record                  {Record of images}
                 ScreenImage : Array[1..4096 ] of Char;
               End;
VAR Ch : Char;
    ColorPage : Array[ 1..4096 ] of Char Absolute $B000 : $0000;
    {This program uses the color screen location.  Change the
    above value to $B000 : $0000 for the monochrome screen.}
    Screen: Array[ 0..3] of StuffInHeap;
    J : Integer;

Procedure WaitForKey;
BEGIN
  REPEAT Read (Kbd,Ch) UNTIL Not KeyPressed;
END;

Procedure Message;  {Asks for keypress for next screen or quit}
BEGIN
  GoToXY(10, 25);
  Write
  ('Press Enter to stop; any other key steps through screens.');
  WaitForKey;
END;

BEGIN {main program}
TextColor(0); TextBackGround(7);
ClrScr;
Write('Screen 0 contains only this line.'); {Write screen 0}
New(Screen[0]);                   {Allocate screen in heapspace}
Move(ColorPage, Screen[0]^.ScreenImage, 4096);  {Store screen }
TextColor(4); TextBackGround(2);      {Attributes for screen 1}
For J := 4 to 15 Do
  BEGIN
    GoToXY(4, J);                           {Write screen 1}
    Write(Chr(186),'Screen 1 is this material.',Chr(186));
  END;
New(Screen[1]);                     {Allocate another screen}
Move(ColorPage, Screen[1]^.ScreenImage, 4096);  {Store screen }
Textcolor(12); TextBackGround(4);
For J := 12 to 22 Do
  BEGIN
    GoToXY(18, J);
    Write
    (Chr(186),'We add this to get screen number 2.',Chr(186));
  END;
New(Screen[2]);
Move(ColorPage, Screen[2]^.ScreenImage, 4096);
TextColor(14); TextBackGround(0);
For J := 15 to 20 Do
  BEGIN
    GoToXY(32, J);
    Write(Chr(186),'And now all of this is screen 3.',Chr(186));
  END;
New(Screen[3]);
{Here we bring screen 3 in from heap:}
Move(ColorPage, Screen[3]^.ScreenImage, 4096);
REPEAT
  Message;             {User presses any key to get next screen}
  If Ch <> ^M Then
    BEGIN              {Bring screen 0 from heap: }
      Move(Screen[0]^.ScreenImage,ColorPage,4096);
      Message;
    END;
  If Ch <> ^M Then
    BEGIN              {Now bring screen 1 from heap: }
      Move(Screen[1]^.ScreenImage, ColorPage, 4096);
      Message;
    END;
  If Ch <> ^M Then
    BEGIN              {Now bring screen 2 from heap: }
      Move(Screen[2]^.ScreenImage, ColorPage, 4096);
      Message;
    END;
  If Ch <> ^M Then     {Now bring screen 3 from heap: }
    Move(Screen[3]^.ScreenImage, ColorPage, 4096);
UNTIL Ch = ^M; ClrScr;
END.
```

A SLEEK NEW STRATEGY FOR

Forward-Looking VDI

The ANSI proposal known as virtual device interface would help end incompatibility among graphics devices by providing a standard graphics interface.

STEVEN ARMBRUST and TED FORGERON



The marvelous richness of function represented by the dozens of new graphics display and input devices released every month is a two-edged sword. Writing a new driver for every display board, printer, or mouse that comes along can get to be an unbearable burden for software developers. Standards are badly needed, but hard to come by—especially in any product niche that is not dominated by IBM.

One such effort at standardization is called VDI (for virtual device interface). This proposed ANSI standard provides a consistent interface to any graphics device. If VDI becomes well-established, "device-driver factories" can produce device drivers to suit almost everyone's needs.

A device driver is the means by which all roughly equivalent devices (printers, video displays) are made to look alike from the standpoint of the applications program (see figure 1). Without device drivers, separate programs would have to be written for each combination of printers, screens, etc.—an impossible task with today's diversity of peripherals. Some farsighted companies have standardized their device drivers across all products they sell, but no such attempt has been made among the many companies in this competitive industry.

VDI defines a common graphics device interface as a collection of subroutines and parameters that describes what the graphics input and output will be. With VDI, a single set of device drivers can be written, and those drivers will work with any programs that adhere to the VDI calling conventions.

IBM released a package in September 1984 called the Graphics Development Toolkit that includes VDI drivers for many IBM-supported devices, including the Professional Graphics Controller, Enhanced Graphics Adapter, Color Graphics Adapter, and several printers. The company that developed those VDI drivers for IBM, Graphic Software Systems of Wilsonville, Oregon, has also released its own package, called GSS-DRIVERS, that includes VDI drivers for many other non-IBM input and output devices.

VDI provides independence for output devices (printers, plotters, monitors) and input devices (mice, tablets, light pens). The applications program uses the same system calls regardless of the device it intends to access. Each VDI driver takes care of translating the system calls into the appropriate action on one particular device. Instead of being limited to just a few devices, an application automatically supports any device for which a VDI driver exists.

VDI is part of a larger set of graphics standards and proposed standards that include GKS (graphical kernel system), VDM (virtual device metafile), and

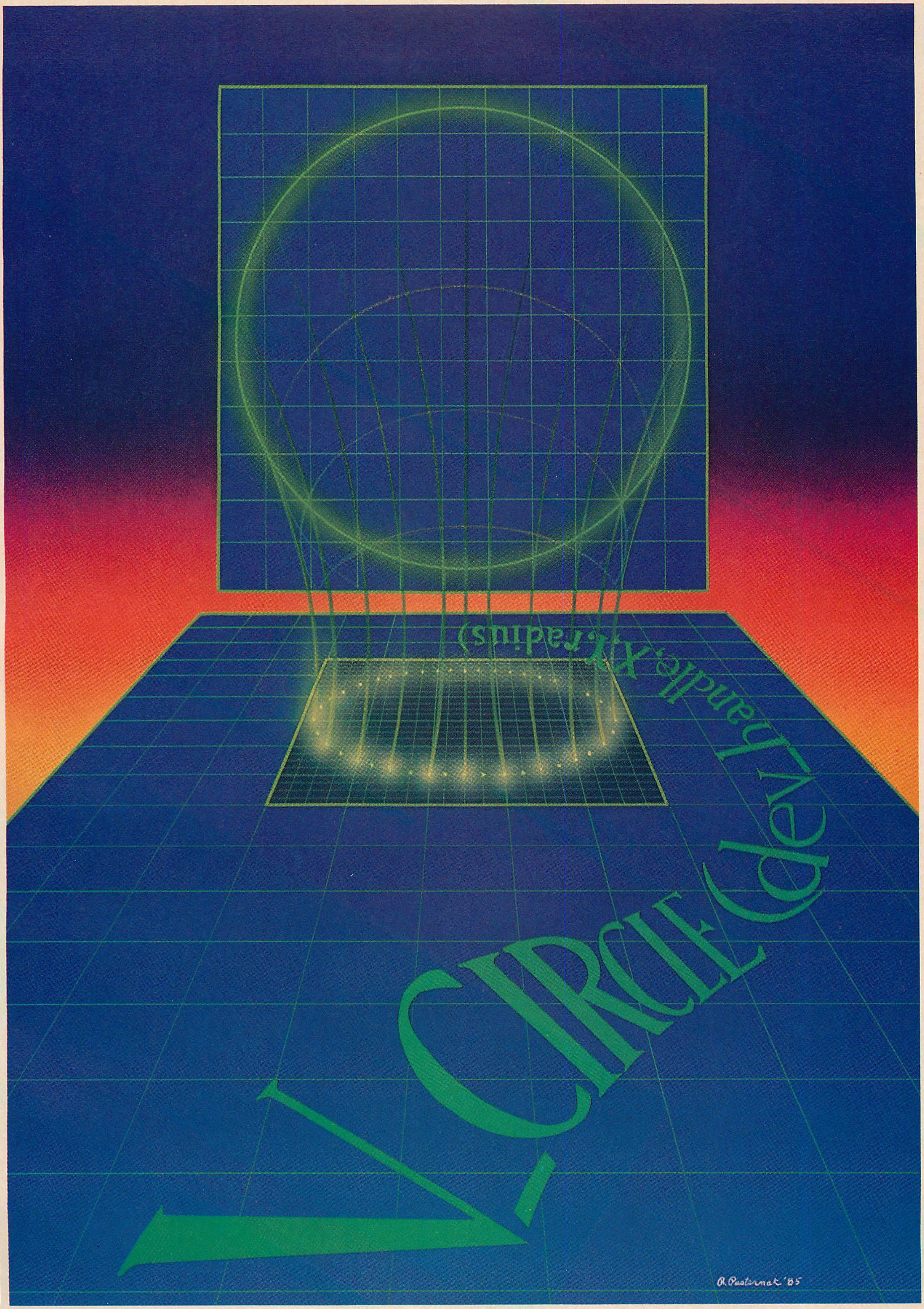


ILLUSTRATION • ROBERT PASTERNACK

NAPLPS (North American presentation level protocol syntax). Figure 2 shows how all of these standards fit together. At the top is GKS, which offers many of the same features as VDI, but at a higher level. Comparing GKS and VDI is like comparing high-level languages and assembly language; GKS offers more conveniences than VDI, but it is slower and requires more memory.

VDM defines a method of storing graphic images on disk and reproducing them later on any VDI-compatible device. NAPLPS defines standards for telecommunicating graphics information to remote terminals, such as videotex terminals that promise shoppers they can examine and purchase merchandise without leaving home.

VDI QUALITY

Besides saving programmer time, VDI can actually improve the quality of output on many devices, because it automatically adjusts output to match the resolution of the device. This can make a great deal of difference even on ordinary dot-matrix printers.

For example, most programs have their own interface to the IBM Graphics Printer. But instead of producing graphic output that matches the resolution of that printer (960 dots across), programs usually dump a screen image from the Color Graphics Adapter (320 dots across). This approach is faster to program, but it is the reason the circle in most pie charts look like a series of jagged steps instead of smooth curves.

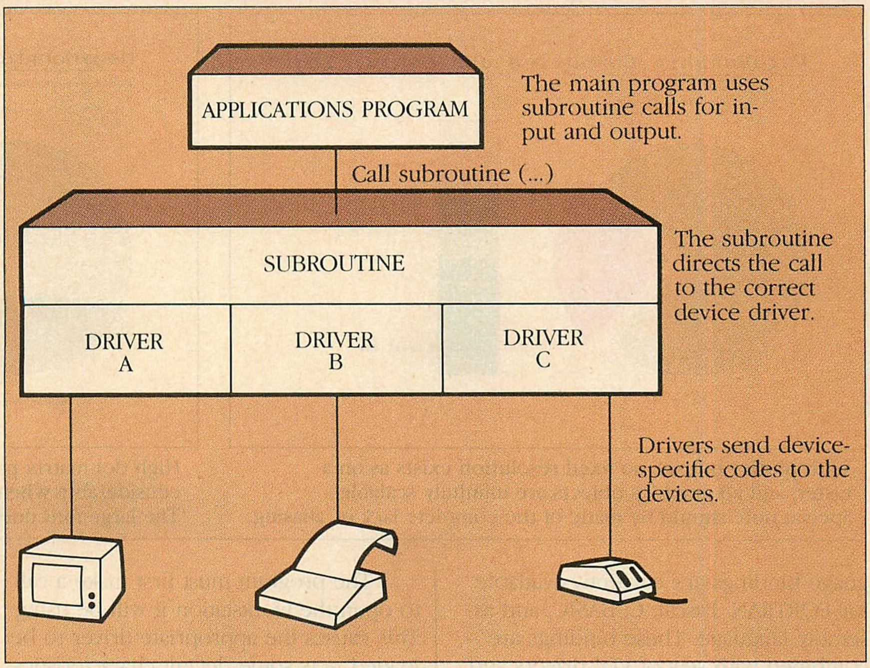
VDI corrects this by using a normalized device coordinate (NDC) space, which has a resolution of 32,768 by 32,768. This resolution is automatically adjusted downward to match that of the device on which the image is displayed. A high-resolution image, complete with color and text of various sizes, retains as much of the detail as is allowed by the device to which it is sent. The same image can be sent to high- or low-resolution devices without changing the program that generated the image.

For example, figures 3 and 4 and photos 1 and 2 show the same graphics image generated by different output devices. A single VDI program (listing 1) produced all of these images.

The characteristics of the images change from device to device, matching the abilities of the individual device (the color devices print in color, the plotter produces smooth curves).

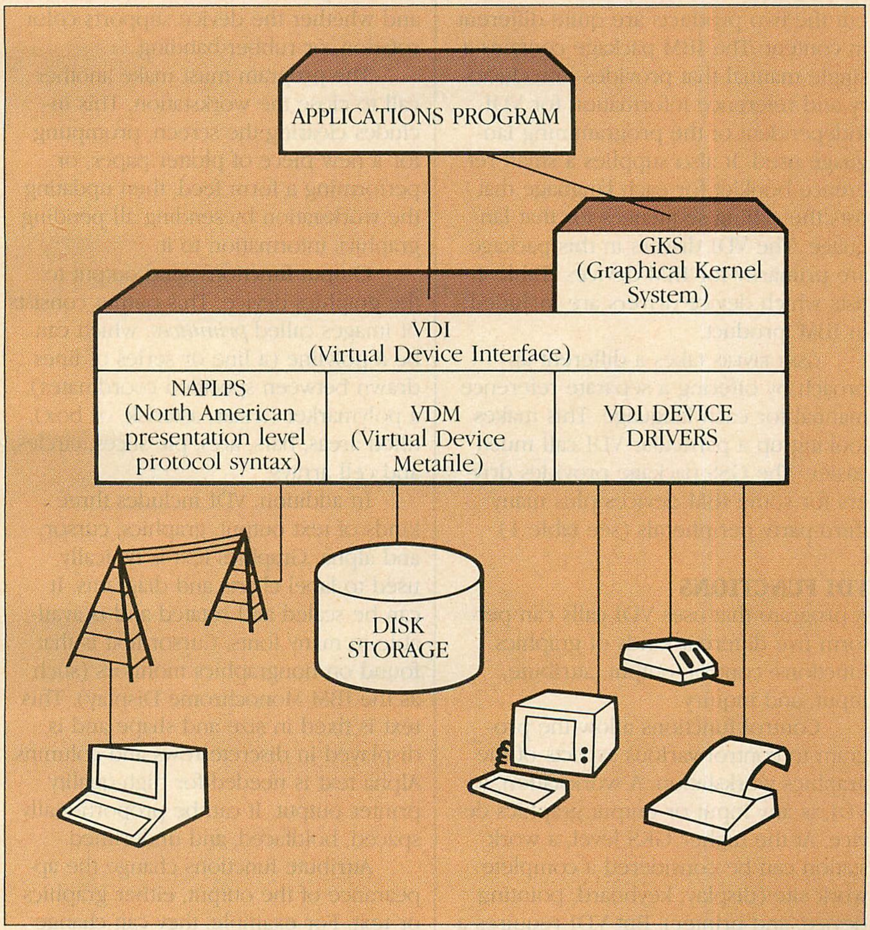
To write applications that use VDI, programmers need language interfaces for the VDI system calls (GSS calls these interfaces *language bindings*). PC lan-

FIGURE 1: *Device Driver/Program Relationship*



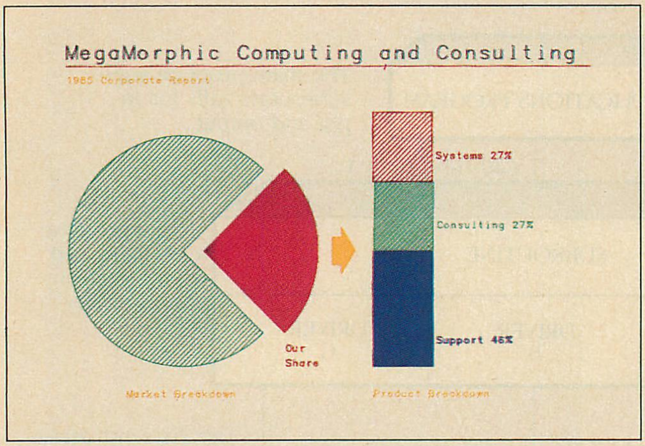
Device drivers are the interface between graphics input and output devices and the main part of the program. They isolate device-specific code.

FIGURE 2: *Graphics Standards*



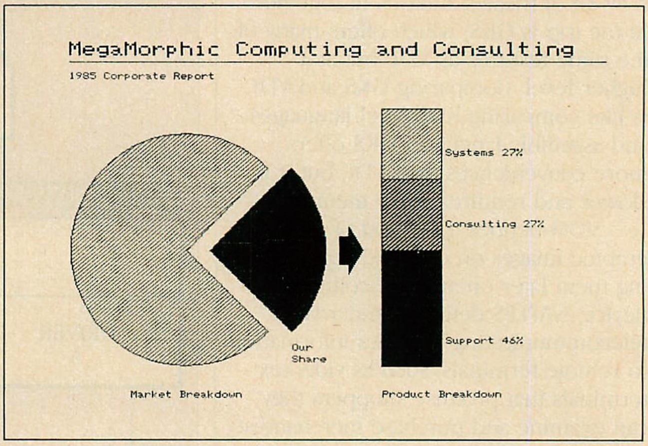
GKS is a higher level interface than VDI, but it is larger and slower. VDI provides access to VDM and NAPLPS, as well as to VDI device drivers.

FIGURE 3: *HP 7475 Output*



With plotter output, no fixed resolution exists as on a raster, and all graphics objects are infinitely scalable. Special note should be made of the complete lack of aliasing.

FIGURE 4: *Epson MX-80 Output*



High dot-matrix printer resolution reduces aliasing considerably when compared to a simple screen dump. The large font does not use the full printer resolution.

guage bindings are currently available for FORTRAN, Pascal, C, BASIC, and assembly language. These bindings are combined with sets of VDI drivers and a programmer's reference manual and are available as development toolkits.

GSS developed the drivers contained in both its own GSS-DRIVERS and IBM's Graphics Development Toolkit, but the two products are quite different in content. The IBM package contains a single manual that provides introductory and reference information for VDI independent of the programming language used. It also supplies a small reference booklet for each language that lists the calling sequences for that language. The VDI drivers in this package are primarily for IBM devices. Table 1 lists which device drivers are included in IBM' product.

GSS-DRIVERS takes a different approach by offering a separate reference manual for each language. This makes looking up a particular VDI call much easier. The GSS package provides drivers for some IBM devices plus many third-party peripherals (see table 1).

VDI FUNCTIONS

A program that uses VDI calls can perform five different kinds of graphics functions: control, output, attribute, input, and inquiry.

Control functions allow the program to control various aspects of the graphics workstation. A workstation, to VDI, is any input or output graphics device. At the higher GKS level, a workstation can be considered a complete work site (display, keyboard, pointing device, and printer). But VDI requires a program to refer to each input and output device as a separate workstation.

The program must first make a call to open the workstation it will be using. This causes the appropriate driver to be loaded, sets some default characteristics, and returns to the program a good deal of information about the device. The information returned includes how high and wide the graphics area is, the number of text fonts and hatching styles, and whether the device supports color, rotation, or rubber-banding.

The program must make another call to close the workstation. This includes clearing the screen, prompting for a new piece of plotter paper, or performing a form feed, then updating the workstation by sending all pending graphics information to it.

Output functions send output to the graphics device. This output consists of images called *primitives*, which can be a polyline (a line or series of lines drawn between specified coordinates), a polymarker (a star, asterisk, or box), filled areas, bars, arcs, pie slices, circles, and cell arrays.

In addition, VDI includes three kinds of text output: graphics, cursor, and alpha. Graphics text is typically used to label charts and diagrams. It can be scaled and rotated and is available in many fonts. Cursor text is that found on nongraphics monitors (such as the IBM Monochrome Display). This text is fixed in size and shape and is displayed in discrete rows and columns. Alpha text is needed for high-quality printer output. It can be proportionally spaced, boldfaced, and underlined.

Attribute functions change the appearance of the output, either graphics or text. For example, they can change the thickness, color, or style (dashed, dotted, solid) of a polyline. They can change the height or rotation of graphics text or the type of fill area.

Input functions return information from a graphics input device, such as a light pen, mouse, or keyboard. These functions operate in two modes: in sample mode, any input from the device is returned to the applications program immediately; in request mode, the user of the input device must signal that the information is ready to be received, for example, by pressing the Enter key.

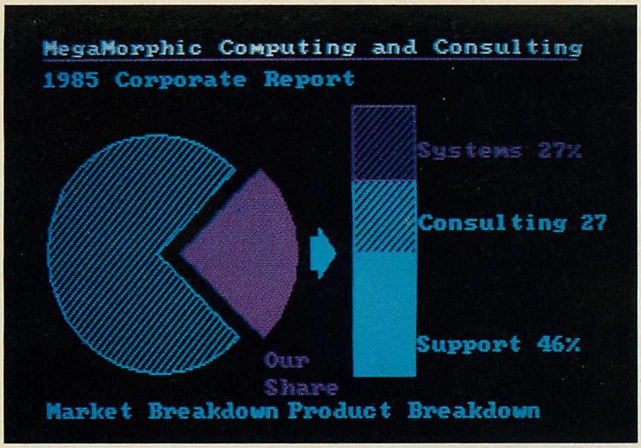
Input functions can be of several types: the locator function returns the current position of the graphics cursor (such as the current position of the mouse); the valuator returns the value of a potentiometer, slide control, or similar input device; the choice function returns a value from a device in which the user has a limited number of choices (such as function keys); string functions return characters typed at the keyboard; cursor movement functions keep track of the arrow keys.

The last type of VDI function, the inquiry function, allows a program to inquire about the current state of affairs and about the workstation's special characteristics. For example, an inquiry function lets a program check the current attributes for polymarkers, polylines, fill areas, etc. The program can check to see if the workstation allows reverse video, if it can generate a hard copy, or if it has raster writing mode.

Generally speaking, each graphics program uses these VDI functions in the following order:

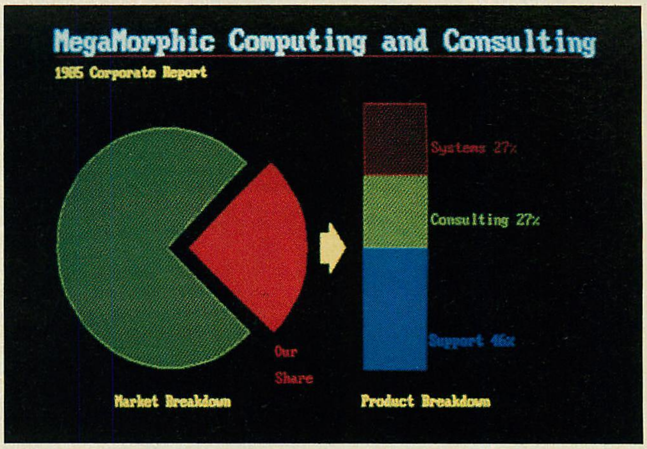
- The program opens the workstation. This prepares the graphics device and returns much information about the device. One of the important decisions a program makes when opening

PHOTO 1: *CGA Display Output*



In order to use color on the IBM Color Graphics Adapter, VDI is limited to 320-by-320 resolution, and, therefore, to 40-column text throughout the graph.

PHOTO 2: *EGA Display Output*



VDI takes advantage of most of the Enhanced Graphics Adapter's advanced features, including the variety of colors, 640-by-350 resolution, and 9-by-14 text font.

the workstation is whether or not to preserve the aspect ratio (the ratio of the horizontal to vertical dimensions of an image). This is important because although VDI's normalized coordinate system allows the same resolution both horizontally and vertically, many devices do not. VDI can display the image to the full resolution of the device both horizontally and vertically (and thus not preserve the aspect ratio). It can preserve the aspect ratio by displaying the image to the resolution of the longest side (possibly clipping off part of the image on the shorter side). For example, if a display screen is rectangular rather than square, and the aspect ratio is not preserved, circles will appear elongated. However, preserving the aspect ratio might clip the top and bottom off large circles.

- After opening the workstation, the program must make sure the device is in the proper mode—either cursor or graphics. These modes are mutually exclusive. In cursor mode, the device is treated like a typical CRT. The program can generate cursor text, erase pages and lines, and control attributes, such as blinking, boldface, and underlining. In graphics mode, the program can produce any graphics output, such as polylines, polymarkers, and circles. It also can set the graphics attributes and produce graphic and alpha text.
- After setting the mode, the program sets the attributes for the graphics output. This might involve setting the character height, line width or type, color, or text type.
- Next, the program sends output to the graphics output device, reads input from the input devices, or inquires about the state of the system.
- When the program is finished, it closes the workstation.

SAMPLE VDI PROGRAM

Listing 1, which produced the output shown in figures 3 and 4 and photos 1 and 2, was compiled with the Lattice C compiler and linked with the GSS-DRIVERS C language bindings. The program consists almost entirely of VDI calls, easily identifiable because they begin with the letter *V*.

The first part of the program sets up data structures used by the rest of the program, mainly by the call to open the workstation. The array WORKIN includes all the input parameters for opening the workstation. These parameters set the beginning conditions for the workstation. This particular array sets the workstation to preserve the aspect ratio, the default polyline to be a solid red line, the default polymarker to be a white star, the default graphics characters to be normal-sized white characters, the default fill characteristics to be green and hollow, and the workstation identifier to be the ASCII characters DISPLAY. DISPLAY provides the workstation's logical name, which is associated with a particular device driver.

Another array, WORKOUT, is set up to receive all the output parameters from setting up the workstation.

The V_OPNWK call opens the workstation. The input parameters are read from array WORKIN. The function returns information about the workstation in array WORKOUT.

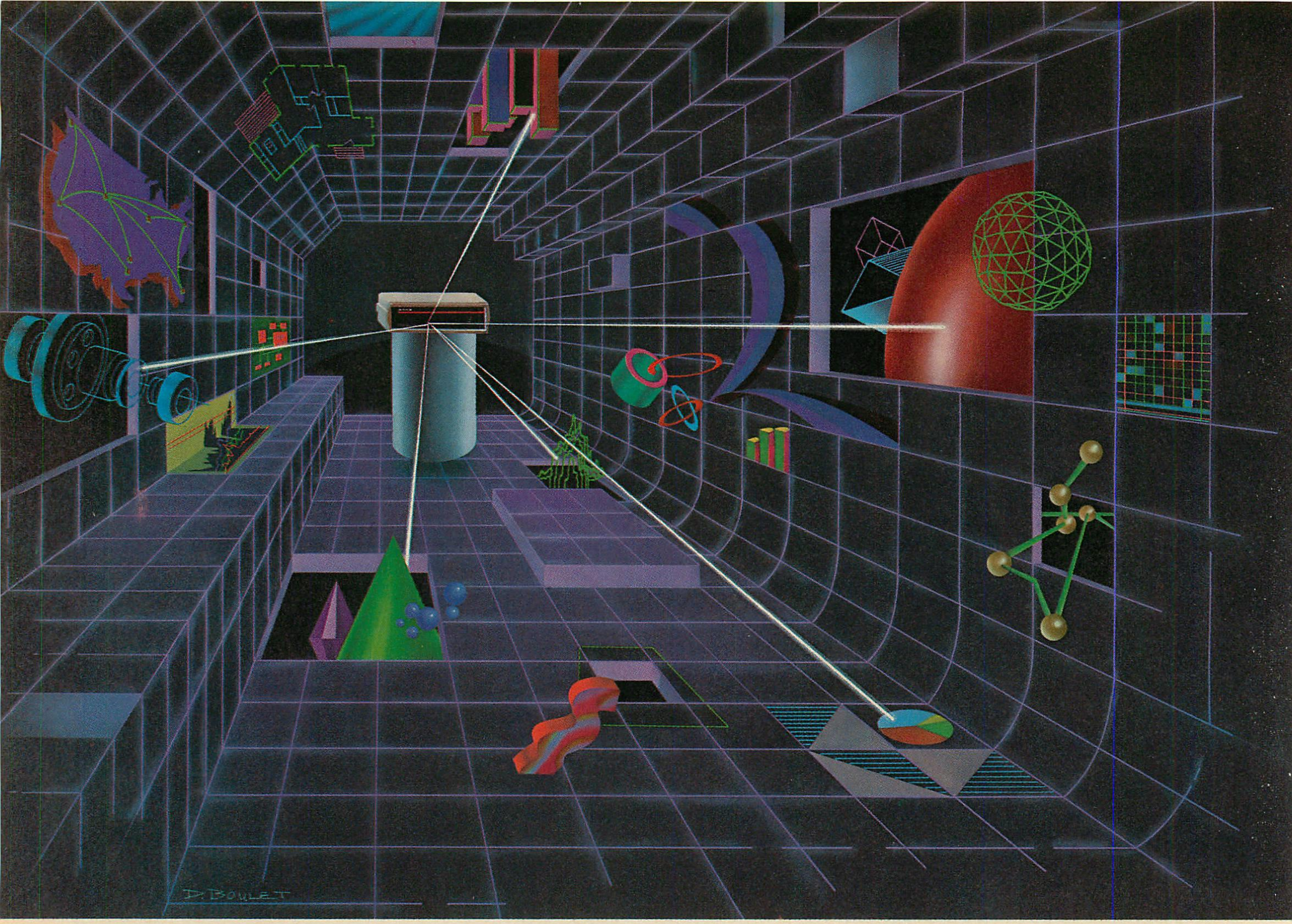
Next, the program uses some of the information in WORKOUT to create an appropriate scaling factor based on the aspect ratio of the device. This scaling factor lets the program make the image as large as possible without clipping any of it. Two of the values that are returned by the V_OPNWK call (WORKOUT[51] and WORKOUT[52]) help determine this factor.

The scaling factor is required because this program makes the assumption that a typical output device has a resolution of 240 by 180 pixels (the program could just as easily have used 600 by 300 or any other set of numbers). Before displaying graphics using VDI, the program must transform this 240-by-180 space into NDC coordinates. The information returned by V_OPNWK is used at this point.

WORKOUT[51] contains the maximum number of NDC coordinates on this device's *X*-axis. WORKOUT[52] contains the maximum on the *Y*-axis. For most monitors these two numbers are unequal, which is why the aspect ratio should be preserved.

The scaling factor for the *X*-axis becomes simply the size of the *X*-axis in NDC units divided by 240 (the size of the *X*-axis in program coordinates). The scaling factor for the *Y*-axis is obtained in a similar manner. To produce a common scaling factor that prevents any part of the image from being clipped, the program uses the smaller of the *X* and *Y* scaling factors and places that value in the variable SCALE. The function TO32K uses this value to transform program coordinates into NDC coordinates.

Next, the program sets the character height for displaying the image's title (*MegaMorphic Computing and Consulting*). This involves a program loop that determines the tallest possible characters that will fit on a single line.


D. BOULET

The program selects a likely character height of 16 pixels (this value is relative to the program's 240-by-180 space). The value is then placed in the variable TITLE_SIZE. The program calls the VDI routine VST_HEIGHT in order to set the character height. One of the values that is returned by VDI_HEIGHT (CELL_WIDTH) indicates the actual width of a character cell in NDC coordinates. With this information, the program can determine if the 36 characters in the title can fit on one line.

The next step is to determine how many NDC units are in a line. That value is placed in LINE_LENGTH. A WHILE loop in the program chooses smaller and smaller character sizes until it calculates that the complete title can fit on a single line.

When the character height is determined, V_GTEXT displays the title in the proper place. The second and third parameters specify the *X*- and *Y*-coordinates of the title's first character.

The program sets the character height to one-half the previous size (by first determining the size and then using VST_ HEIGHT to set it). The alignment of the text is changed so that when the program displays more text, it will indicate text position by specifying the top of the leftmost character (the default is the bottom of the leftmost character). VST_COLOR changes the text color to yellow.

Then, V_GTEXT writes out the second line of text. The solid line between the two lines of text is achieved by drawing a polyline. The array XY holds the *X* and *Y* coordinates, and the call to V_PLINE draws the line. VST_COLOR changes the text color to red, and two calls to V_GTEXT write the words *Our Share*, one beneath the other. VST_ALIGNMENT and VST_COLOR change the alignment and the color of the text once more, and V_GTEXT writes the three labels for the bar chart.

Next, the program draws the bar chart itself. VSF_INTERIOR sets the interior style to solid, VSF_COLOR sets the fill color to blue, and V_BAR draws the bar within the coordinates specified in the array XY. To outline the bar, VSF_INTERIOR sets the interior style to empty, and V_BAR draws the outline. Similar calls are used to produce the rest of the bar graph, changing the fill style and color as appropriate.

After writing the subtitles for the pie graph and bar graph, the program draws the pie graph itself. V_PIESLICE draws the pie, beginning with the large slice. As with the bar chart, this routine is called twice for each slice: once to fill in the interior and once to outline the figure. The first two parameters of V_PIESLICE indicate the center of the circle. The third indicates the radius. The last two indicate the starting and ending angles of the slice in tenths of degrees (in this case, from 45 to 135 degrees). To offset the small slice from the rest of the pie, a different *X*-coordinate is used for the center of the circle.

The arrow is the last element to be drawn. V_FILLAREA draws it and fills it in, much like V_PIESLICE and V_BAR. The coordinates for the arrow are defined in the array ARROW and transformed and placed into array XY.

The next VDI call, VRQ_STRING, causes the program to wait until the user types a character string and presses Enter. Otherwise, the program would exit to DOS and erase the graphic image before the user had time to examine the screen. If the first character of the input string is *C*, the program makes another VDI call, V_HARDCOPY, to send a copy of the image to the printer.

For its final clean-up operations, the program calls V_CLRWK to clear the image from the workstation (and do a form feed or page eject if the workstation is a hard-copy device). Then it calls V_ENTER_CUR to leave graphics mode and enter cursor mode. This call is advisable if the program is to display output at the IBM Color Graphics Monitor. Otherwise, the display will be left in 40-column mode. The program calls V_CLSWK to close the workstation.

The output of this program will vary among different devices. For example, on some devices the first line may be in large characters, while on others it will be the same size as the rest of the characters. This is because many devices are able to display characters in only a few discrete sizes. Regardless of the size specified by the program, the VDI driver for each device translates the requested character size into the device's closest supported character size.

Other variations may be manifested in colors displayed. The Hewlett-Packard plotter (figure 3) is able to display all the colors requested by the program, while the IBM Color Graphics Adapter is more limited. As with character size, the drivers select the closest supported colors.

INSTALLING DEVICE DRIVERS

Before the program in listing 1 can be run, the appropriate VDI drivers have to be installed. First, the CONFIG.SYS file is modified. That file contains the names of the installable device drivers that get loaded when DOS starts running. DEVICE commands are added to this file for all the VDI drivers that are needed. To produce output on six different devices, for example, commands for six different device drivers would have to be included.

The following commands specify device drivers for the Epson MX-80 printer, the NEC 3550 printer, the HP 7475 plotter, the IBM Enhanced Graphics Adapter, the IBM Color Graphics Adapter (high-resolution, black-and-white graphics), and the Color Graphics Adapter (medium resolution, black-and-white graphics).

```
DEVICE=EPMX80.SYS /g:output
DEVICE=NEC3550.SYS /g:output
DEVICE=HP7475.SYS /g:output
DEVICE=VDIDY010.SYS /g:output
DEVICE=IBMBW.SYS /g:output
DEVICE=IBMCO.SYS /g:output
```

The /g option at the end of these commands indicates that they are all part of the same group—called output. Drivers in the same group share memory. That is, VDI reserves enough memory for the largest driver in the group, and the drivers are swapped in and out of memory depending on which device the graphics program opens. Because drivers in a group share memory, only one device can be open at a time.

Another option for VDI drivers, the /r option, specifies that the driver is always resident in memory. Devices with this option can be opened at the same time. The command

```
DEVICE=GSSVDI.SYS
```

loads the resident portion of VDI. This command must appear after all other DEVICE commands for VDI drivers.

With this CONFIG.SYS file, when the computer starts running, the resident portion of VDI is loaded, along with each resident driver and, by convention, the last driver in each group.

The DRIVERS command must then be entered (or included in the AUTOEXEC.BAT file). This command informs the operating system that the resident portion of VDI is loaded and available for graphics operations.

If the executable version of the program shown in listing 1 is in a file called VDIGRAPH.EXE, the program can be run by typing VDIGRAPH. This would send the output, in color, to the display connected to the Color Graphics Adapter, because, in this case, that was the last device listed.

To produce output on a different device, use the DOS SET command to change the driver associated with the

name DISPLAY. (Remember, the program opens the logical device named DISPLAY.) For example, to produce output on the Epson MX-80 type:

SET DISPLAY = EPMX80.SYS

Then invoke the program; the output would go directly to the MX-80.

The SET command can be used to direct the output of this program to any of the devices for which there are drivers listed in the CONFIG.SYS file. It also can be used to send the output to a different port. For example, the command

SET EPMX80.SYS = LPT2

would send the output to an Epson MX-80 connected to printer port LPT2.

One consideration in using VDI is the amount of memory required. Table 1 lists the memory sizes for various VDI drivers—both IBM and GSS versions. Sizes for GSS version 1.0 and the newer version 1.03 are given when those sizes differ. The resident portion of VDI (GSSVDI.SYS in the GSS version or VDI.SYS in the IBM version) requires approximately 32KB of memory. Individual drivers require from 2 to 30KB, depending on how the CONFIG.SYS file is set up. If all the drivers are part of the same group, just one block of memory big enough to hold the largest driver is required. Other drivers are loaded into this memory area as needed.

If the CONFIG.SYS file is set up to keep multiple drivers in memory at the same time, performance increases because VDI does not need to load drivers every time a new device is used. But this scheme requires more memory dedicated to VDI. In general, the resident portion of VDI, together with a printer driver and a display driver (both sharing the same memory), require approximately 64KB.

PERFORMANCE PROBLEMS

VDI has been criticized for its performance, which must improve before it can be considered a PC standard. VDI builds a graphics image in a 32K-by-32K NDC space and maps that image down to the resolution of a specific device; that process takes a long time.

Screen output should be crisp and snappy, especially for word processors, windowing programs, and drawing packages, all of which are prime candidates for VDI use. Nothing leaves a worse impression than a program that displays output to the screen as if the data were traveling across the country via a 1200-baud modem.

Both the IBM and GSS VDI display drivers have noticeable performance

TABLE 1: *Memory Requirement*

PERIPHERAL	IBM DRIVER NAME	IBM DRIVER MEMORY	GSS DRIVER NAME	GSS DRIVER MEMORY
IBM Game Adapter and PC*jr* Joystick	VDIGIJOY.SYS	2,720	—	—
IBM CGA and EGA (high-resolution, b/w)	VDIDY006.SYS	16,816	IBMBW.SYS	14,192/ 17,648
IBM CGA and EGA (medium-resolution, color)	VDIDY004.SYS	17,792	IBMCO.SYS	15,152/ 19,392
IBM EGA (monochrome monitor)	VDIDY00F.SYS	19,200	—	—
IBM EGA (enhanced color display)	VDIDY010.SYS	19,680	—	—
IBM EGA (medium-resolution, 16-color)	VDIDY00D.SYS	19,632	—	—
IBM EGA (high-resolution, 16-color)	VDIDY00E.SYS	19,632	—	—
IBM Compact Printer	VDIPRCOM.SYS	20,256	—	—
IBM Color Printer	VDIPRCOL.SYS	25,792	—	—
IBM Graphics Printer	VDIPRCOL.SYS	20,176	IBMGRP.SYS	21,376
Houston Instrument DMP-29 Plotter	—	—	HIPT29.SYS	10,192
IDS Prism Printers (b/w, 8-inch wide)	—	—	IDS80BW.SYS	19,024
IDS Prism Printers (color, 8-inch wide)	—	—	IDS80CLR.SYS	21,684
IDS Prism Printers (b/w, 13-inch wide)	—	—	IDS132BW.SYS	19,456
IDS Prism Printers (color, 13-inch wide)	—	—	IDS132CL.SYS	23,104
Koala Pad Touch Tablet	—	—	KOALA.SYS	2,400
NEC 7730 letter-quality Printer	—	—	NEC7730.SYS	22,080
NEC 3550 letter-quality Printer	—	—	NEC3550.SYS	22,384
Nicolet Zeta 8 Plotter	—	—	ZETAB.SYS	10,032
Okidata 93 and 84 Printers	—	—	OKID84.SYS	22,736
Okidata 92 Printer	—	—	OKID92.SYS	20,512
Printronix MVP, P300, and P600 Printers	—	—	PRNTRX.SYS	24,480
IBM PC*jr* (high-resolution, 4-color)	VDIDY00A.SYS	17,120	—	—
IBM PC*jr* (medium-resolution, 16-color)	VDIDY009.SYS	17,424	—	—
IBM PC*jr* (low-resolution, 16-color)	VDIDY008.SYS	17,440	—	—
IBM 7372 Color Plotter	VDIPLSIX.SYS	12,896	—	—
IBM 7371 Color	VDIPLTWO.SYS	11,696	—	—

PERIPHERAL	IBM DRIVER NAME	IBM DRIVER MEMORY	GSS DRIVER NAME	GSS DRIVER MEMORY
Virtual Device Metafile	VDIMTFIL.SYS	16,864	META.SYS	16,864
Amdek AMPLOT II Plotter	—	—	AMPLOT.SYS	9,504
Data South 180 Printer	—	—	DS180.SYS	24,432
Diablo 150C Ink-jet Color Printer	—	—	DIAB150.SYS	28,672/ 36,160
Epson MX-100 Printer	—	—	EPMX100.SYS	21,472
Epson MX-80 Printer with Graphtrax Plus	—	—	EPMX80.SYS	21,616
Hewlett-Packard 7470A Plotter	—	—	HP7470.SYS	11,696
Hewlett-Packard 7475A Plotter	—	—	HP7475.SYS	12,896
Hewlett-Packard HP2225C Printer	—	—	HP2225.SYS	20,160
QUME Sprint 11 Plus Printer	—	—	QUME11.SYS	22,528
Seikosha GP-700A Color Printer	—	—	GK700.SYS	21,424
Strobe Plotters	—	—	STROBE.SYS	10,400
Summagraphics Summamouse	—	—	SUMMAMS.SYS	2,208
Summagraphics Summatablet	—	—	SUMMATB.SYS	2,176
VDI Controller (required)	VDI.SYS	31,504	GSSVDI.SYS	31,520

This table lists the VDI drivers in the IBM Graphics Development Toolkit and GSS-DRIVERS packages. It also lists the memory these drivers require. If a size difference exists between versions 1.0 and 1.03 of the GSS drivers, both sizes are listed. If no values or names are listed, a driver for that device is not included in the package.

TABLE 2: *VDI Performance*

BENCHMARK	IBM CGA High-Res., B/W	IBM CGA Medium-Res., Color
10 PAGES GRAPHICS TEXT		
IBM 1.0	79	145
GSS 1.0	79	145
GSS 1.03	12	23
10 PAGES CURSOR TEXT		
IBM 1.0	27	27
GSS 1.0	27	27
GSS 1.03	25	25
DRAW SAMPLE GRAPH		
IBM 1.0	6	3
GSS 1.0	6	3
GSS 1.03	3	2

All times are shown in seconds.

The first three rows of numbers were obtained by timing the program in listing 2. The cursor text figures were obtained from timing the program in listing 3. The times required to draw the graph of listing 1 are shown in the final three rows.

problems. For example, the IBM package required six seconds to display the output from the program in listing 1 on a Color Graphics Adapter (using the high-resolution, black-and-white driver).

With the release of GSS-DRIVERS 1.03 in April 1985, GSS took a giant step forward in solving the slow-performance problem. The display drivers in 1.03 performed 2 to 10 times faster than their 1.0 counterparts. IBM has not yet updated its VDI drivers to incorporate the same GSS breakthroughs.

GSS has also increased the speed of displaying graphics text, as the numbers in table 2 indicate. The first three rows show the speed of graphics text using two different drivers: the high-resolution, black-and-white and the medium-resolution, color drivers for the Color Graphics Adapter. These numbers were obtained by running the program shown in listing 2 (VDIGTEXT) on a standard IBM PC with an 8088 processor. The program simply goes into graphics mode and displays 10 pages of text, 80 characters per line.

Even though GSS has made great strides toward improving the performance of graphics display output, VDI's cursor text is as slow as ever. The times listed in table 2 were obtained by running VDICTEXT (listing 3). This program displays the same data as the program in listing 2, using cursor text instead of graphics text.

These figures do not compare favorably with those obtained by programs that do not use VDI. For example, when the same 10 pages of data used in listings 1 and 2 were typed into a file, the IBM Personal Editor took less than 3 seconds to display them, including the time for the operator to press the PgDn key 10 times. VDI took 25 to 27 seconds to display the cursor text.

One reason that VDI's cursor text is so much slower than programs without VDI is that VDI drivers perform cursor text output via the ROM BIOS. Most popular programs skip the BIOS and send output directly to video RAM. In this way, programs achieve sharp, snappy screen output, but they also run the risk of future hardware incompatibility. When IBM provides multitasking support in DOS, programs that do not use the BIOS for screen output likely will be unable to run. To avoid this, GSS chose the safe approach with the VDI drivers, forsaking some degree of performance in the bargain.

The future is bright for VDI, in spite of its current performance problems. Many manufacturers have already incorporated VDI into their products.

VisiCorp pioneered the use of VDI drivers in VisiOn. Other companies, such as Ashton-Tate and Microrim, are committed to using VDI drivers in their products. Many PC developers, however, are waiting for certain questions to be answered before jumping on the VDI bandwagon. Besides performance, these questions involve the ability to get new drivers into the hands of enough people to make VDI worthwhile.

Although the drivers in the IBM and GSS packages support many peripheral devices, and GSS plans to add more drivers for popular devices, new devices are being introduced all the time. How will VDI drivers for these new devices be written?

One way is for the manufacturers or ISVs to write the drivers themselves. IBM has published the specifications for VDI driver support. Peripheral manufacturers can contract with GSS to have the drivers written. GSS charges for the service, but it is expert in the field and has the channels of distribution to get new drivers out to the public.

GSS also is working on a language bindings writer's kit, which will allow vendors of high-level languages to create their own bindings for VDI. As more high-level language products include VDI language bindings, commercial software products will be more likely to support VDI.

Even if compiler manufacturers include VDI language bindings, and developers write their applications to support VDI, the people who purchase the applications have no guarantee they can take advantage of the VDI support. In order to benefit from VDI, users must have their own copies of VDI drivers for the devices they own.

Currently, users have three ways to obtain copies of VDI drivers. First, the drivers could come bundled with an applications program. For a $100 registration fee and a $500-per-application, unlimited licence fee, software developers can include IBM's VDI drivers with the applications they sell. GSS offers a similar arrangement. The licence fees are not unreasonable, but the cost of three extra diskettes (which increases as more drivers are written) per copy of an application can be prohibitive in a competitive marketplace.

Second, users can buy VDI drivers directly. In mid-1985, GSS plans to distribute its drivers (not the full toolkit, just the drivers) through retail outlets. It plans to sell three different diskettes, each of which will contain at least 12 drivers. One disk will have all of the most popular drivers, while the other two disks will contain drivers for less-frequently-used devices. Each of these disks will sell for about $30 and will include telephone support.

The third, and perhaps best, approach is for hardware manufacturers to distribute VDI drivers for the devices they manufacture. This is frequently done today with non-VDI device drivers for realtime clocks and nonstandard disk-drive products. Buying the driver along with the hardware device prevents the customer from paying for disks full of drivers he will never use.

Eventually, as VDI is implemented in silicon, it may migrate into the devices themselves. Then, instead of requiring extra memory in the computer for the VDI drivers, the resident portion of VDI could send VDI calls directly to the devices for interpretation. This could go a long way toward correcting VDI's size and speed problems.

VDI is a forward-looking concept. In the past, when computers were simpler and available peripherals fewer in number, VDI was not necessary, and the slower, memory-limited computers could not afford to use it. The availability of increased computer power realized with faster, stronger CPUs, such as the iAPX286 and 386, should make VDI practical just about the time that the ever-exploding array of graphics peripherals makes VDI necessary.

GSS-DRIVERS: $200
Graphic Software Systems, Inc.
25117 Southwest Parkway
Wilsonville, OR 97070
503/682-1606
CIRCLE 302 ON READER SERVICE CARD

Graphics Development Toolkit: $350
IBM
P.O. Box 1328
Boca Raton, FL 33132
800/447-4700
CIRCLE 303 ON READER SERVICE CARD

Steven Armbrust is a freelance technical writer. Ted Forgeron is a microcomputer software consultant. They work primarily in the "Silicon Forest" west of Portland, OR.

LISTING 1: VDIGRAPH.C

```
/-
 *  Program to do the MegaMorphic Computing and Consulting
 *  corporate report chart for 1985 using VDI.
 */

#define MIN(A,B) (A)<(B) ? (A) : (B)

#define BLACK 0
#define WHITE 1
#define RED 2
#define GREEN 3
#define BLUE 4
#define YELLOW 5
#define CYAN 6
#define MAGENTA 7

int scale;

main()
{
int xy[100],
    i,
    height_req,
    line_length,
    char_width,
    cell_width,
    cell_height,
    title_size,
    hor_out,
    workout[66],
    device_handle,
    vert_out;

static  int   workin[] = {1,1,RED,3,1,1,WHITE,0,0,GREEN,
                          1,'D','I','S','P','L','A','Y',' '};
extern  int   scale;
static  int   echo_xy[] = {0, 0};
static  int   arrow[] = {130, 90, 136, 80, 130, 70,
                         130, 74, 126, 74, 126, 86,
                         130, 86, 130, 90};
char    input_string[2];

    /* initialize the device and put characteristics in workout */
    v_opnwk(workin,&device_handle,workout);

    /* create scaling based on aspect ratio of the device */
    scale = MIN(workout[51] / 240, workout[52] / 180);

    /* set a new character height */
    title_size = 16;
    vst_height(device_handle,to32k(title_size),
               &char_width,&cell_width,&cell_height);
 /* find out how long ruling is in NDC */
 line_length = to32k(240);

 /*
  * loop making sure that the title won't be
  * longer than the ruling
```

```
*/
while (cell_width >= line_length  / 36 &&
      title_size >= 2) {

    /* try a new height */
    vst_height(device_handle,
                   to32k(--title_size),&char_width,
                   &cell_width,&cell_height);
}

/* write out title */
v_gtext(device_handle,to32k(10),to32k(163),
             "MegaMorphic Computing and Consulting");

/* compute size for other text */
height_req = (float)to32k(title_size) / 2;

/* set character height smaller */
vst_height(device_handle,height_req,
          &char_width,&cell_width,&cell_height);

/* set connecting point to upper left corner of text */
vst_alignment(device_handle,0,2,&hor_out,&vert_out);

/* change text color */
vst_color(device_handle,YELLOW);

/* write out subtitle */
v_gtext(device_handle,to32k(10),to32k(156),
             "1985 Corporate Report");

/* create data to draw ruling */
xy[0] = to32k(10); xy[1] = to32k(160);
xy[2] = to32k(240); xy[3] = to32k(160);

/* draw the ruling */
v_pline(device_handle,2,xy);

/* change text color */
vst_color(device_handle,RED);

/* write out pie label */
v_gtext(device_handle,to32k(106),to32k(40),
             "Our");
v_gtext(device_handle,to32k(106),
             (int)(to32k(40)-cell_height*1.5),
             "Share");
/* set alignment to bottom left */
vst_alignment(device_handle,0,0,&hor_out,&vert_out);

/* change text color */
vst_color(device_handle,BLUE);

/* write out labels for bar chart */
v_gtext(device_handle,to32k(172),to32k(40),
             "Support 46%");

vst_color(device_handle,GREEN);
v_gtext(device_handle,to32k(172),to32k(90),
             "Consulting 27%");

vst_color(device_handle,RED);
v_gtext(device_handle,to32k(172),to32k(120),
             "Systems 27%");

/* set the interior style to solid */
vsf_interior(device_handle,1);

/* create coordinates for first bar */
xy[0] = to32k(144); xy[1] = to32k(30);
xy[2] = to32k(170); xy[3] = to32k(80);

/* change fill color */
vsf_color(device_handle,BLUE);

/* draw bar */
v_bar(device_handle,xy);

/* set the interior style to empty */
vsf_interior(device_handle,0);

/* outline the bar */
v_bar(device_handle,xy);

/* modify for second bar */
xy[1] = to32k(80); xy[3] = to32k(110);

/* change fill color */
vsf_color(device_handle,GREEN);

/* set interior style to hatch */
vsf_interior(device_handle,3);

/* use narrow 45 degree lines */
vsf_style(device_handle,1);

/* draw second bar */
v_bar(device_handle,xy);

/* set the interior style to empty */
vsf_interior(device_handle,0);

/* outline the bar */
v_bar(device_handle,xy);

/* modify for third bar */
xy[1] = to32k(110); xy[3] = to32k(140);

/* change fill color */
vsf_color(device_handle,RED);

/* set interior style to hatch */
vsf_interior(device_handle,3);

/* use medium 45 degree lines */
vsf_style(device_handle,2);

/* draw third bar */
v_bar(device_handle,xy);

/* set interior style to empty */
vsf_interior(device_handle,0);

/* outline the bar */
v_bar(device_handle,xy);

/* set text alignment to top center */
vst_alignment(device_handle,1,2,&hor_out,&vert_out);

/* change text color */
vst_color(device_handle,YELLOW);

/* write out subtitles */
v_gtext(device_handle,to32k(60),to32k(20),
             "Market Breakdown");
v_gtext(device_handle,to32k(170),to32k(20),
             "Product Breakdown");

/* change fill color */
vsf_color(device_handle,GREEN);

/* set fill interior to hatch */
vsf_interior(device_handle,3);

/* draw pie slice */
v_pieslice(device_handle,to32k(60),to32k(80),
             to32k(50),450,3150);

/* set interior style to empty */
vsf_interior(device_handle,0);

/* draw the outline of the big pie slice */
v_pieslice(device_handle,to32k(60),to32k(80),
             to32k(50),450,3150);

/* change fill color */
vsf_color(device_handle,RED);

/* set the interior style to solid */
vsf_interior(device_handle,1);

/* draw small pie slice */
v_pieslice(device_handle,to32k(70),to32k(80),
             to32k(50),3150,450);
```

```
/* set interior style to empty */
vsf_interior(device_handle,0);

/* outline the small pie slice */
v_pieslice(device_handle,to32k(70),to32k(80),
                to32k(50),3150,450);

/* change fill color */
vsf_color(device_handle,YELLOW);

/* set interior style to solid */
vsf_interior(device_handle,1);

/* for every one of the points in the arrow array */
for (i=0;i<=15;i++)     {

  /* transform the points */
  xy[i] = to32k(arrow[i]);

 }

/* fill the arrow */
v_fillarea(device_handle,8,xy);

/* set the interior style to empty */
vsf_interior(device_handle,0);

/* outline the arrow */
v_fillarea(device_handle,8,xy);

/* wait for <CR> */
vrq_string(device_handle,2,0,echo_xy,input_string);

/* if first char is upper c do a hardcopy */
if (input_string[0] == 'C')     {

  /* call for hardcopy */
  v_hardcopy(device_handle);
}
    /* clear the workstation */
    v_clrwk(device_handle);

    /* leave graphics mode */
    v_enter_cur(device_handle);

    /* close down the workstation */
    v_clswk(device_handle);

}

int   to32k(world)
int   world;

/*
 *  function to take world coordinates
 *  and translate them to NDC space.
 */

{
    extern  scale;

    return(world * scale);
}
```

LISTING 2: VDIGTEXT.C

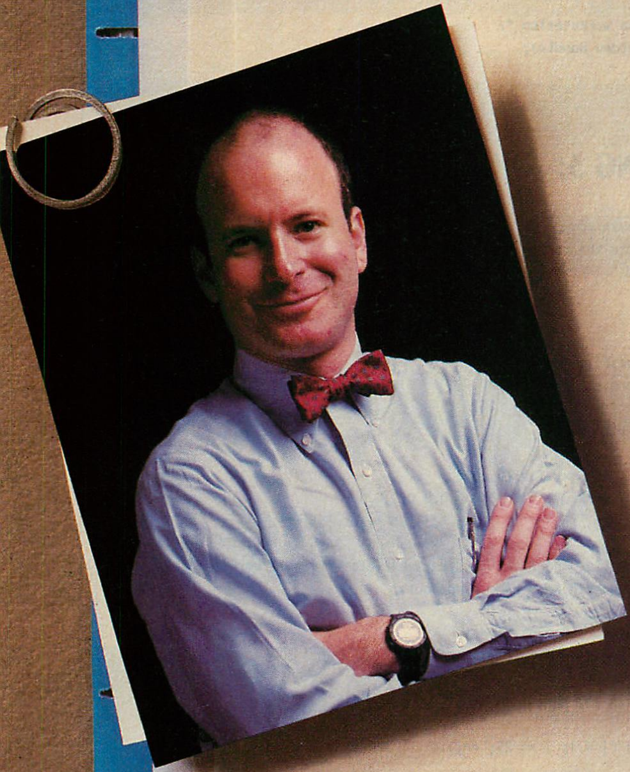
```
/*
 *  Program to output 10 pages of graphics
 *  text using the VDI graphics text feature.
 */

main()
{
int  workout[66];
int  dev_handle;
int  i,j;
char input_string[2];

static int   echo_xy[] = {0, 0};
static int   workin[]  = {1,1,1,1,1,1,1,1,1,1,
                         1,'D','I','S','P','L','A','Y',' '};

    /* initialize the device and put characteristics in workout */
    v_opnwk(workin,&dev_handle,workout);

    for (i = 1; i <= 10; ++i) {

        for (j = 1; j <= 25; ++j) {

            /* output graphics text */
            v_gtext(dev_handle,1,j*911,
                    "This is a test of the VDI graphics text.");
            v_gtext(dev_handle,16384,j*911,
                    "This is a test of the VDI graphics text.");
        }

        /* clear the screen */
        v_clrwk(dev_handle);
    }

    /* close workstation */
    v_clswk(dev_handle);

}
```

LISTING 3: VDICTEXT.C

```
/*
 *  Program to output 10 pages of text
 *  using the VDI cursor text features.
 */

main()
{
int  workout[66];
int  dev_handle;
int  i,j;
char input_string[2];

static int   echo_xy[] = {0, 0};
static int   workin[] = {0,0,0,0,0,0,0,0,0,0,
                        0,'D','I','S','P','L','A','Y',' '};

    /* initialize the device and put characteristics in workout */
    v_opnwk(workin,&dev_handle,workout);

    /* enter cursor text mode */
    v_enter_cur(dev_handle);

    for (i = 1; i <= 10; ++i) {

        for (j = 1; j <= 25; ++j) {

            /* specify direct cursor address */
            vs_curaddress(dev_handle,j,1);

            /* output cursor addressable text */
            v_curtext(dev_handle,
                    "This is a test of the VDI cursor text.  ");
            v_curtext(dev_handle,
                    "This is a test of the VDI cursor text.  ");
        }

        /* move cursor home */
        v_curhome(dev_handle);

        /* erase to end of screen */
        v_eeos(dev_handle);
    }

    /* close workstation */
    v_clswk(dev_handle);

}
```

C O M P A N Y   M I S S I O N

His mission is to streamline communications by making different brands of micros talk to each other in the same network.



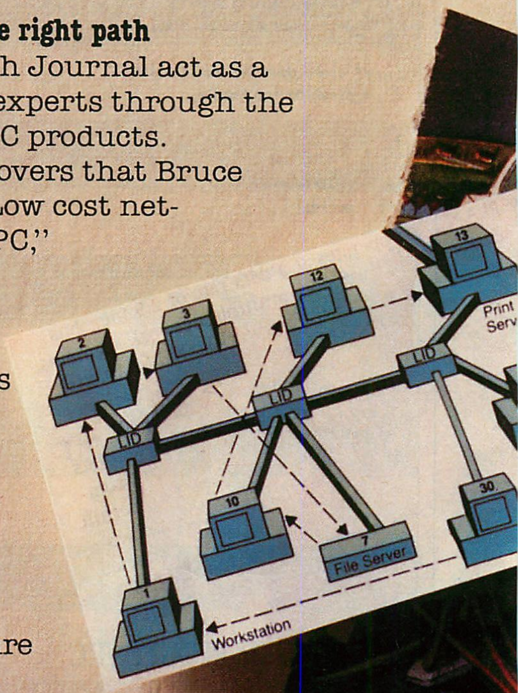
He's just one of 75,000 advanced PC experts who uses PC Tech Journal to fulfill his professional responsibilities as a problem-solver. His recommendation is more often than not the last word in his company's computer purchases.

When one person has this much influence, he has to know more about state-of-the-art PC technology than anyone else in his company. That's why he reads PC Tech Journal.

Clues that put him on the right path

The articles in PC Tech Journal act as a compass to direct PC experts through the thickening forest of PC products.

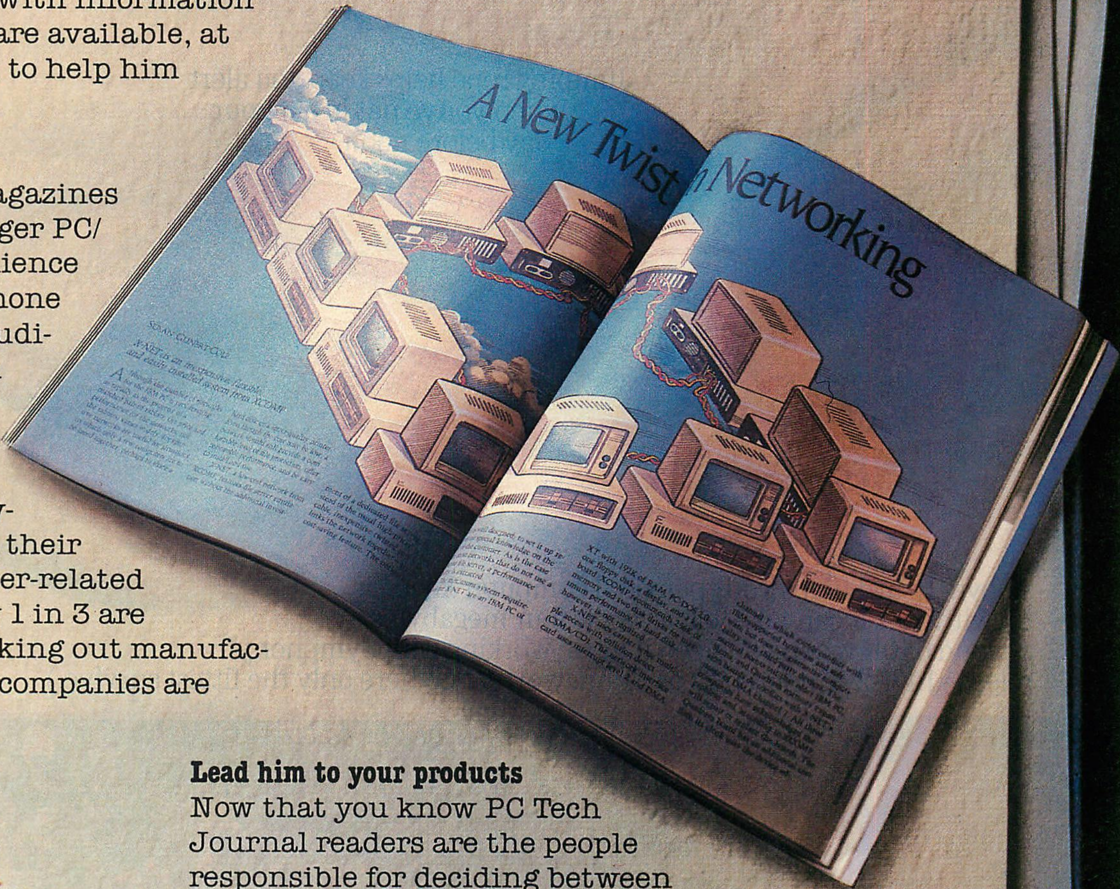
This PC expert discovers that Bruce Churchill's article, "Low cost networking for the IBM PC," points to many of the networking products he will need. And William Colsher's tips on sharing files keep him from wandering into the valley of lost data.



Clues that help him perform his best

It's not only the feature articles that give him direction, though.

The ads, too, keep him on course. They provide him with information on what products are available, at this very moment, to help him complete his task.



Stay hot on his trail

There are other magazines that give you a larger PC/PC compatible audience of IBM users, but none that give you an audience with so much buying power.

Eight out of ten PC Tech Journal power buyers are involved in their company's computer-related purchases. (Nearly 1 in 3 are responsible for seeking out manufacturers when their companies are acquiring LANs).

Lead him to your products

Now that you know PC Tech Journal readers are the people responsible for deciding between your product and a competitor's, reach them before the next decision is made.

Make it your mission to influence the power buyers who influence their companies. It's easy! Just call your PC Tech Journal sales representative, now, and make PC Tech Journal his guide to your products.

PC TECH JOURNAL

Unmatched power in your market

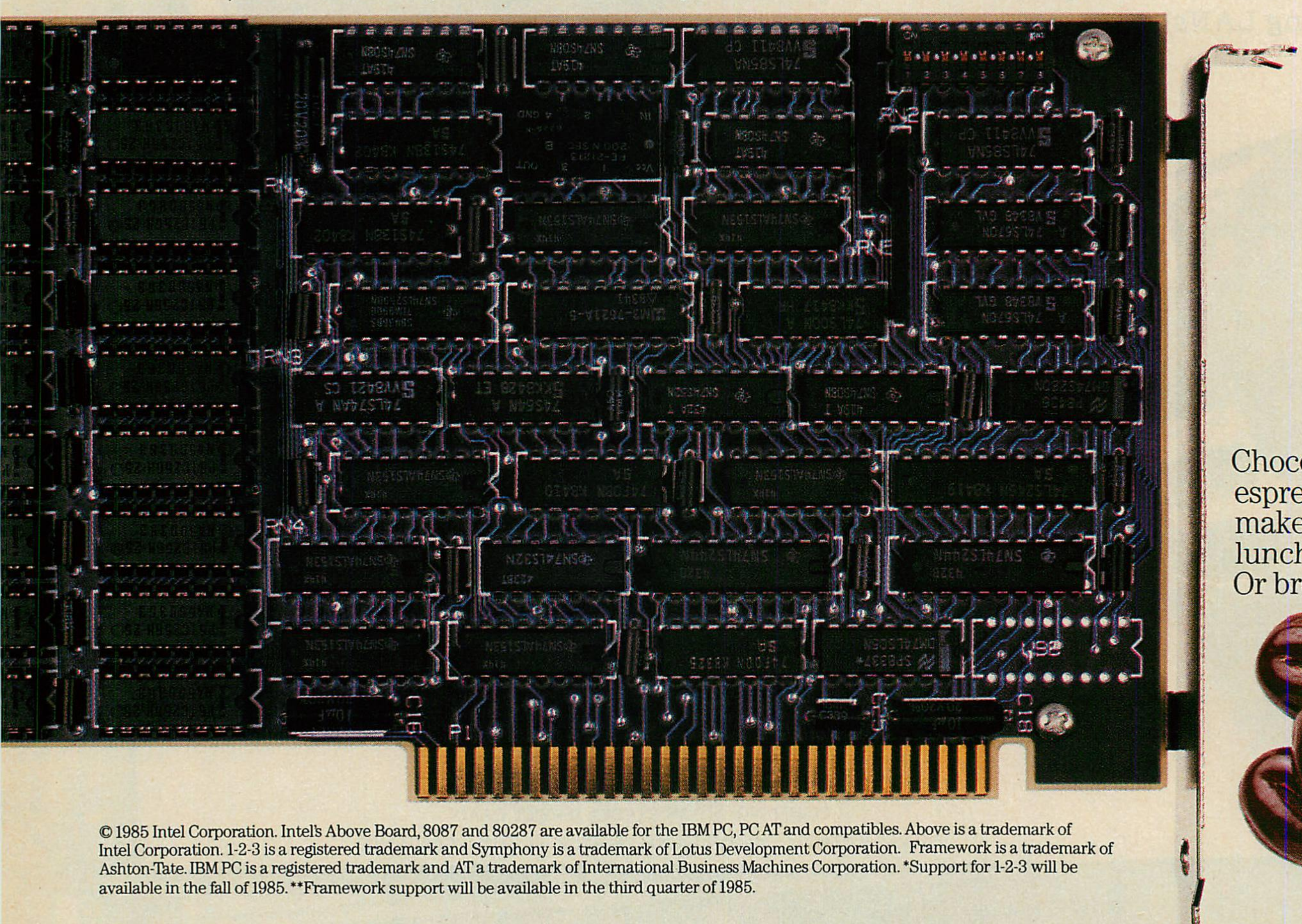
CIRCLE NO. 246 ON READER SERVICE CARD

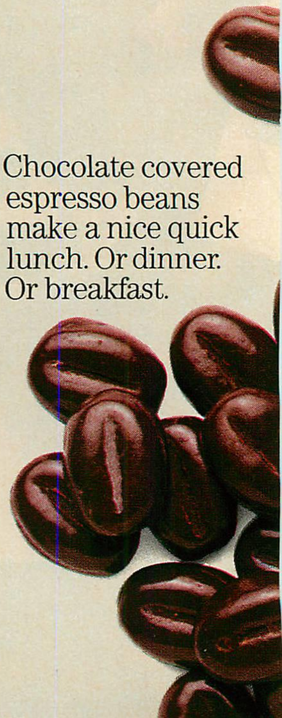
Ziff-Davis Publishing, One Park Avenue, New York, NY 10016 (212) 503-5100

Survival kit for

Jumping rope helps keep you alert when you've put in a 6 hour day before noon.

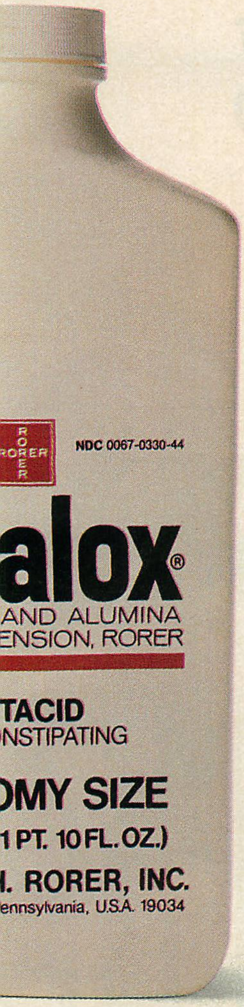
Above™ Board helps your personal computer keep up with you. It can take you as far as four megabytes beyond 640K when you need it most: Working with Symphony™ release 1.1, 1-2-3® release 2,* or Framework.™** (Where only the fittest survive.)





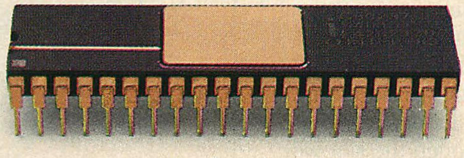
Chocolate covered espresso beans make a nice quick lunch. Or dinner. Or breakfast.

overachievers.

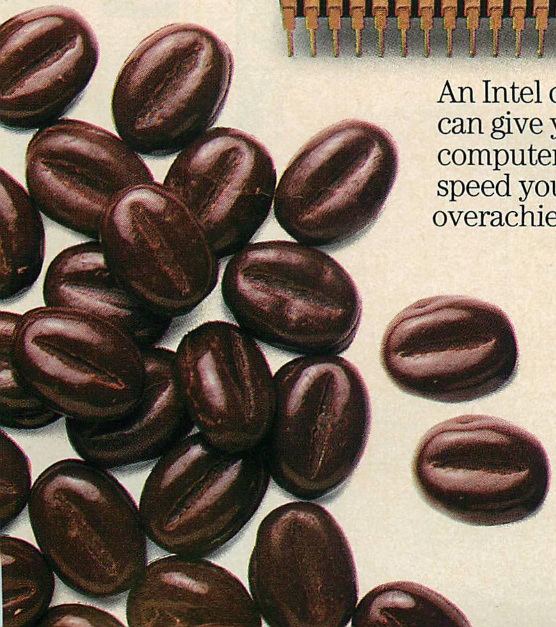


Two old sweaters for when you're working late and you can see your breath because they've turned off the heat in the office.

When you get into Q4 on your spreadsheet and run out of memory, Maalox can help soothe your stomach.



An Intel coprocessor can give your personal computer the extra speed you need to keep overachieving.



Life isn't easy when you're overworked and underappreciated. That's why Intel created Above Board. It's a new kind of memory board designed to meet the specification jointly developed by Intel and Lotus Development Corporation.

Above Board does all the things you'd expect any self-respecting memory board to do. But because it lets your personal computer work so much harder, you can put off such uncomfortable things as cutting and pasting. Or such disruptive things as converting up to a mini.

You can also get an Intel coprocessor to give you the extra speed you need to keep up with all that new memory. Intel has coprocessors that will let your IBM PC® or PC AT™ run up to five times faster. Or more.

You can see and touch and buy these Intel products at your favorite computer store. Or call 800-538-3373 for the names of dealers near you.

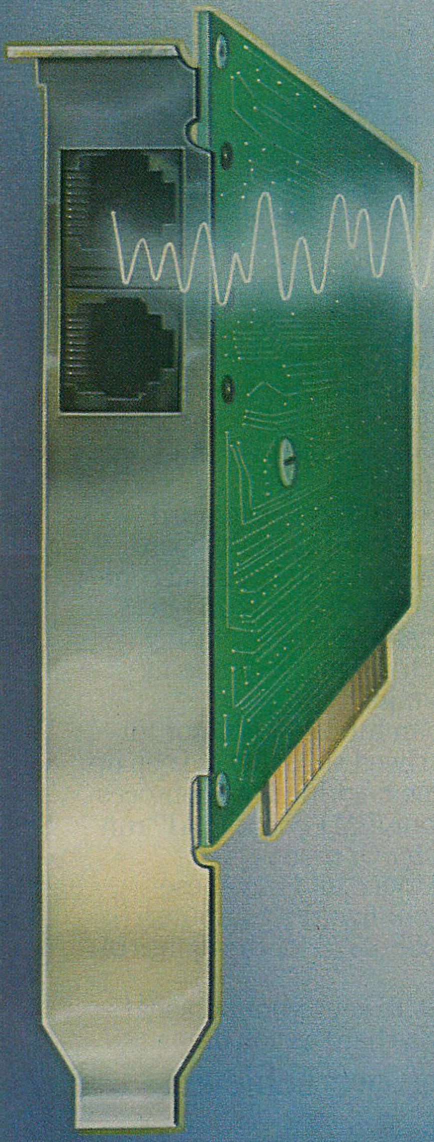
And find out how to keep your personal computer working as hard as you do.

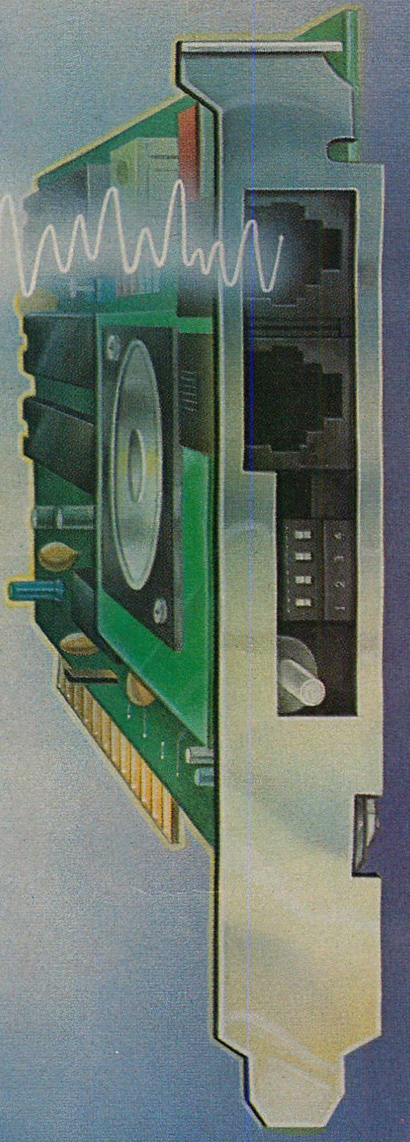
Help for the overachiever.

intel®

CIRCLE NO. 216 ON READER SERVICE CARD

Communicating from Within



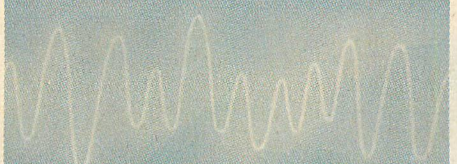


The debate over internal versus external modems will probably never end. Each type has valid arguments in its favor. Those who advocate external modems over the internal variety claim a lack of flexibility in the internal arrangement; a modem made to fit inside an IBM PC is not going to hook up easily to another type of computer. They point out that setting switches and controlling speaker volume are usually easier on an external modem. They also like the flashing front panel lights so they can monitor the progress of a call. Furthermore, they argue, internal modems use up an expansion slot. Although the slot will also be used with an external modem, which requires a serial port, most serial ports are mounted on a board with four or five other functions. A computer with an internal modem would require an additional slot for these extra functions.

The major arguments put forth for an internal modem involve its efficient (or nonexistent) use of desk space and its lack of an unsightly external power supply with the attendant cable and

Internal modems may be out of sight but never out of mind, as we review 13 of them available for the PC family.

AUGIE HANSEN



clutter. Internal-modem advocates point to the ease of transporting the modem along with the computer. Where security is an issue, an internal modem is more difficult to steal.

Internal modems came into being after the external units had become established, and, perhaps for that reason, have had to work harder to prove themselves. They may not be as widely used as external modems, but the market supports a number of them. For those who decide that an internal modem is the right answer for them, this article examines 13 such products for the PC family. (For a helpful introduction to modems, see the article "Modems Demystified," David Schwaderer, *PC Tech Journal*, July 1984, p. 73.)

ILLUSTRATION • ANDY LEVINE

EVALUATION CRITERIA

The process of evaluating these 13 modems and their associated software packages was tedious but enlightening. The criteria used in this evaluation are compatibility, performance, hardware and software features, product and documentation quality, and ease of installation, set-up, and operation. Each product was set up for COM2, installed in a PC, and tested using the supplied software; it was then tested again in a PC/AT. The operational tests included routine and worst-case data calls, tests of terminal emulation and file-transfer protocol features, and normal computer activity (reading the mail, programming, and documentation tasks).

Table 1 summarizes the products. Suggested retail prices range from $399 to $549. A two-year limited warranty is the most popular offering. Microcom's Era 2 has a four-year limited warranty.

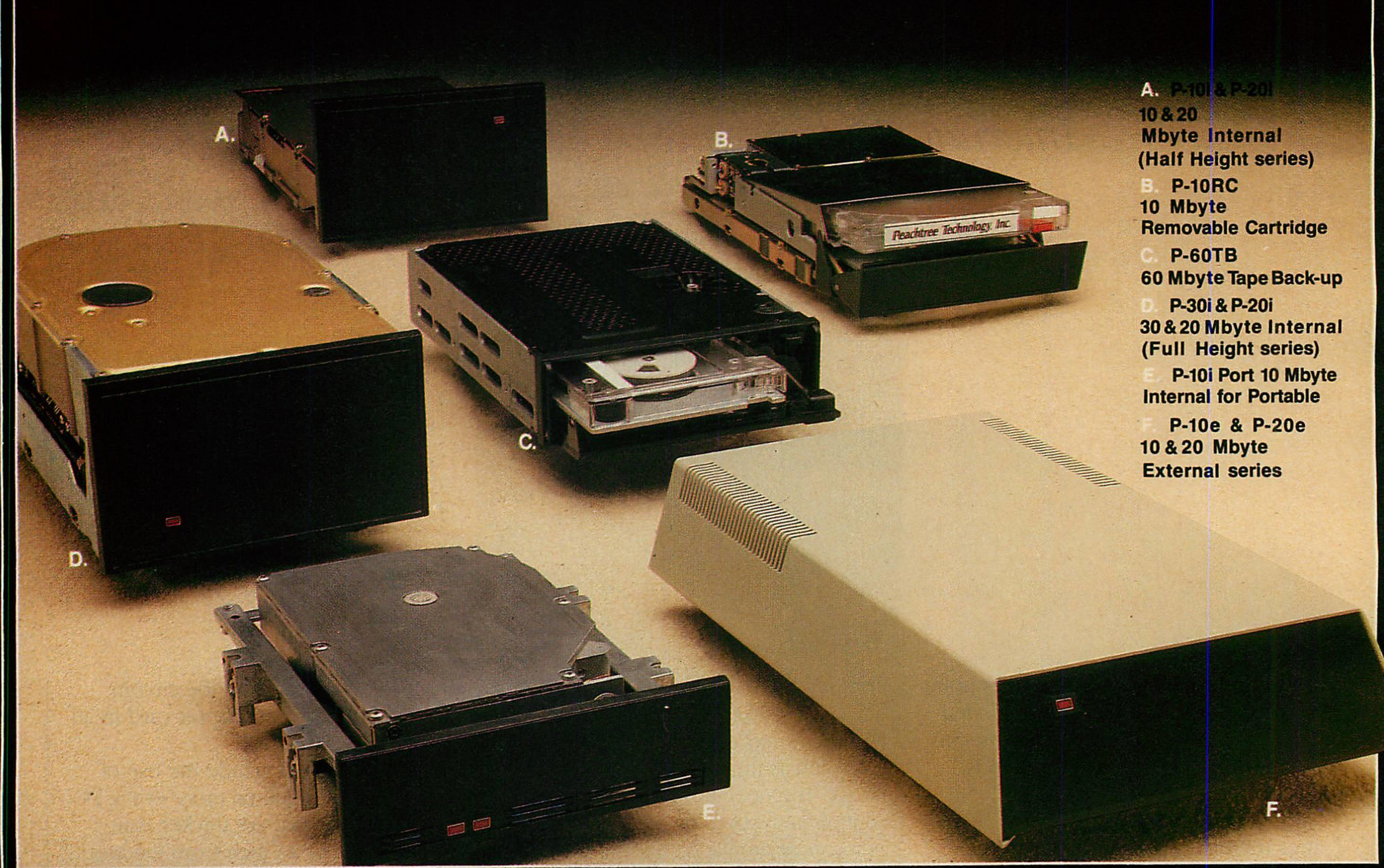
Figure 1 compares the physical profiles of all the modems. Four of them are built on half-sized cards, permitting their use in the PC, PC/XT, Portable PC, and PC/AT because they can fit in the narrow, short slots. Unfortunately, the U.S. Robotics board has a descending skirt with a circuit path that can possibly short pins in the AT bus if it is plugged in anywhere except slots 1 or 7. The AST Reach! board has a metal tab sticking up from one corner of the speaker that must be bent down flat in order to avoid touching the circuit paths on an adjacent adapter card in an XT, Portable PC, or AT.

Two factors preclude the use of some boards in one or more of the PC models. Those that are thicker than three-fourths of an inch from the base of the circuit board, including protruding wires or components, to the top of the tallest component will not fit comfortably in XTs, Portables, and ATs because these machines have narrow board-to-board spacing. Boards that have descending skirts along the connector edge of the board will probably not sit properly in an AT because of the bus expansion connector. Only slots 1 and 7 can accept such boards, which include those produced by Microcom, Anderson Jacobson, and PROMETHEUS. The RIXON modem has a partially descending skirt that appears to be unnecessary but is just deep enough to prevent the card's use in all but the AT's two special PC-compatible slots.

Table 2 is a comparison of features implemented in hardware and firmware. All of the modems can handle

The Drive Warriors

Restoring order to the mass memory market!



The mass memory market has become a real battle zone! Chaos, confusion and buyer uncertainty have resulted from too many products and not enough answers.

The truth of the matter is, anyone can sell a winchester disk drive. The problem is that hard drives, by their own nature, are sensitive devices that require appropriate handling and customer support.

Don't be mislead into believing that a disk drive can be sold and booted around the country, without any risk.

There is no room for the weak, when it comes to delivering a product you can depend on!

Peachtree Technology Inc. has cleared the path of confusion with a comprehensive line of mass memory expansion systems that have established themselves as leaders in advanced hard drive technology, backed by strong customer support. Every Peachtree Technology Inc. product has been through stringent diagnostic testing as a complete system. To assure they not only meet, but surpass specifications. (On the outside of every box is a documented certificate of inspection signed by our technician). Plus we stand behind our products with a complete one year parts & labor warranty.

FLEXIBILITY AND COMPATIBILITY A MUST!

To eliminate frustration, Peachtree Technology Inc. has provided a complete line of mass memory storage systems. They are fully compatible with IBM PC, XT, & PORTABLE, AT&T, CORONA PORTABLE, COMPAQ DESK PRO & PORTABLE, COLUMBIA, TAVA, ZENITH 150 AND MANY MORE. (If your computer is not listed, please contact us for further information).

We offer a complete line of products including:

- External winchester hard drives • 10, 20, and 30 megabyte (formatted capacity). Our 10 and 30 megabyte drives utilize a hard carbon coating on the disk platter surface.
- Internal mounted winchester hard drives • 10, 20 and 30 megabyte (formatted capacity)
- Removable hard disk • Half height winchester, 10 megabyte, utilizing new and advanced head positioning technology.
- Cartridge tapesystems • Capacity – up to 60 mbytes • Half high 5.25″ form factor add-in/add-on to IBM-PC and compatibles.
- A 135 watt internal power supply that replaces the standard unit in the PC.

We'll supply everything you need to give your micro new speed and storage capacity: Disk drive, controller, cables, software, and associated custom firmware.

One stop shopping for your mass memory requirements.

Don't become a victim! The frustration and inconvenience of not having customer support may not be worth the few dollars you save buying from an unreliable source. There is no substitute for a complete hands on demonstration and face to face commitment to customer service from a reputable dealer. So don't be left standing in the smoke. To find out more about Peachtree Technology Inc. and our complete line of memory expansion systems, ask your local computer dealer for a complete demonstration. If he does not have our products have him call 404-662-5556, and we will deliver a demonstration unit & literature within 24 hours!

CIRCLE NO. 158 ON READER SERVICE CARD

Peachtree Technology, Inc.

3020 Business Park Drive, Norcross, Georgia 30071, Telephone (404) 662-5158 Sales (404) 662-5556

110-, 300-, and 1200-bps full- or half-duplex operation. All can use pulse or tone dialing or a combination of the two for special circumstances.

The industry's de facto standard, the Hayes Smartmodem (see sidebar), permits reversal of the transmitted and received frequencies in order to allow the modem to connect with a remote originate-only modem. All but two of the internal modems tested do the same. The Smartmodem uses an analog loopback self-test to verify correct operation of its internal circuitry. Ten of the 13 modems provide some form of local analog loopback, remote digital loopback (which tests external lines in addition to internal circuitry), or both for self-testing purposes.

Most of the modems have a built-in speaker for audio monitoring of a call's progress. Modem commands allow users to turn the speaker off entirely, leave it on all the time, or monitor the line until a connection is established. The last option is the most popular. It permits the user to hear the dial tone, outpulsing or tone dialing, acquisition tone from the distant end, and the start of data transmission tones before the sound is turned off.

The Smart-Cat PLUS does not have a built-in speaker, but instead uses a cable to feed its signal to the PC's speaker. The Racal-Vadic and RIXON offerings make no provision for a speaker; the only feedback they provide is a display of running commentary on the progress of a call, but with no specifics. The status of the call can be determined only by picking up the telephone.

All of the modems work with the RJ11 connection, a two-wire feed that services a standard (nonelectronic) telephone. Only five—IntelliModem, POPCOM, ProModem, Smart-Cat PLUS and Smartmodem—utilize an RJ12 or RJ13 connector. They use the A-lead control of what are called multiline feeds, a four-wire arrangement that permits signaling for the control of busy lamps when at least one of the connected telephone sets is active. Figure 2 shows these circuit arrangements schematically.

A telephone set may be installed with the modem in one of two ways. A duplex jack may be used to place the telephone in parallel with the modem, or the telephone may be placed in series with the modem by plugging it into the phone jack provided on all of the modems (figure 3).

The Hayes Smartmodem opens the circuit to the telephone when the modem is activated. This permits switching from voice to data without having to

TABLE 1: *Product Summary*

MODEM	VENDOR ADDRESS	PRICE	WARRANTY	COMPATIBILITY
AJ Connection	Anderson Jacobson, Inc. 521 Charcot Avenue San Jose, CA 95131 408/263-8520	$495 $544 with asynchronous adapter	2-year	PC/AT–slots 1 and 7 only
Encore 1200B	OmniTel 3090 Oakmead Village Drive Santa Clara, CA 95051 408/986-8236	$399 $449 with optional aux. port	2-year	
Era 2	Microcom, Inc. 1400A Providence Highway Norwood, MA 02062 617/762-9310	$499	4-year	Requires two adjacent slots; PC/AT–slot 1 only
IntelliModem XT	Bizcomp 532 Mercury Drive Sunnyvale, CA 94086 408/733-7800	$549	2-year	
Maxwell 1200PC	Racal-Vadic 1525 McCarthy Blvd. Sunnyvale, CA 94088-3442 408/946-2227	$495	2-year	
Microlink	U.S. Robotics 1123 W. Washington Blvd. Chicago, IL 60607 312/733-1053	$499	2-year	PC/AT–slots 1 and 7 only
PC212A	RIXON, Inc. 8310 Guilford Road Columbia, MD 21046 301/381-2300	$499	2-year	PC/AT–slots 1 and 7 only
PC Modem Half Card	Ven-Tel, Inc. 2342 Walsh Avenue Santa Clara, CA 95051 408/727-5721	$549	2-year	
POPCOM C100	Prentice Corp. 266 Caspian Drive Sunnyvale, CA 94088-3544 408/734-9810	$445	2-year	
ProModem 1200B	PROMETHEUS Products 4545 Cushing Parkway Fremont, CA 94530 415/490-2370	$399	1-year	Requires two adjacent slots; PC/AT–slot 1 only
Reach!	AST Research, Inc. 2121 Alton Avenue Irvine, CA 92714 714/863-1333	$549	1-year; $50 for 1-year extension	
Smart-Cat PLUS	Novation, Inc. 20409 Prairie Street Chatsworth, CA 91311 818/996-5060	$449 $399 without software	2-year	
Smartmodem 1200B	Hayes Microcomputer Products, Inc. P.O. Box 105203 Atlanta, GA 30348 404/441-1617	$549	2-year	

Suggested retail prices span the range from $399 to $549. Some modems have an auxiliary serial port. A two-year limited warranty is the most popular offering. All can be used in the PC, XT, Portable PC, and AT unless otherwise noted.

hang up and redial, and it prevents an active data connection from being disturbed when someone takes the telephone off-hook. Modems by AST, Microcom, OmniTel, and PROMETHEUS work the same way. All the others leave the connection live, which means the telephone is effectively connected in parallel. Prentice considers this to be a feature and uses the off-hook condition of the series-connected telephone as a signal to drop the data connection. If the telephone is on a parallel tap, the data connection is not dropped, although it becomes garbled, as it does with any other modem that is parallel-connected.

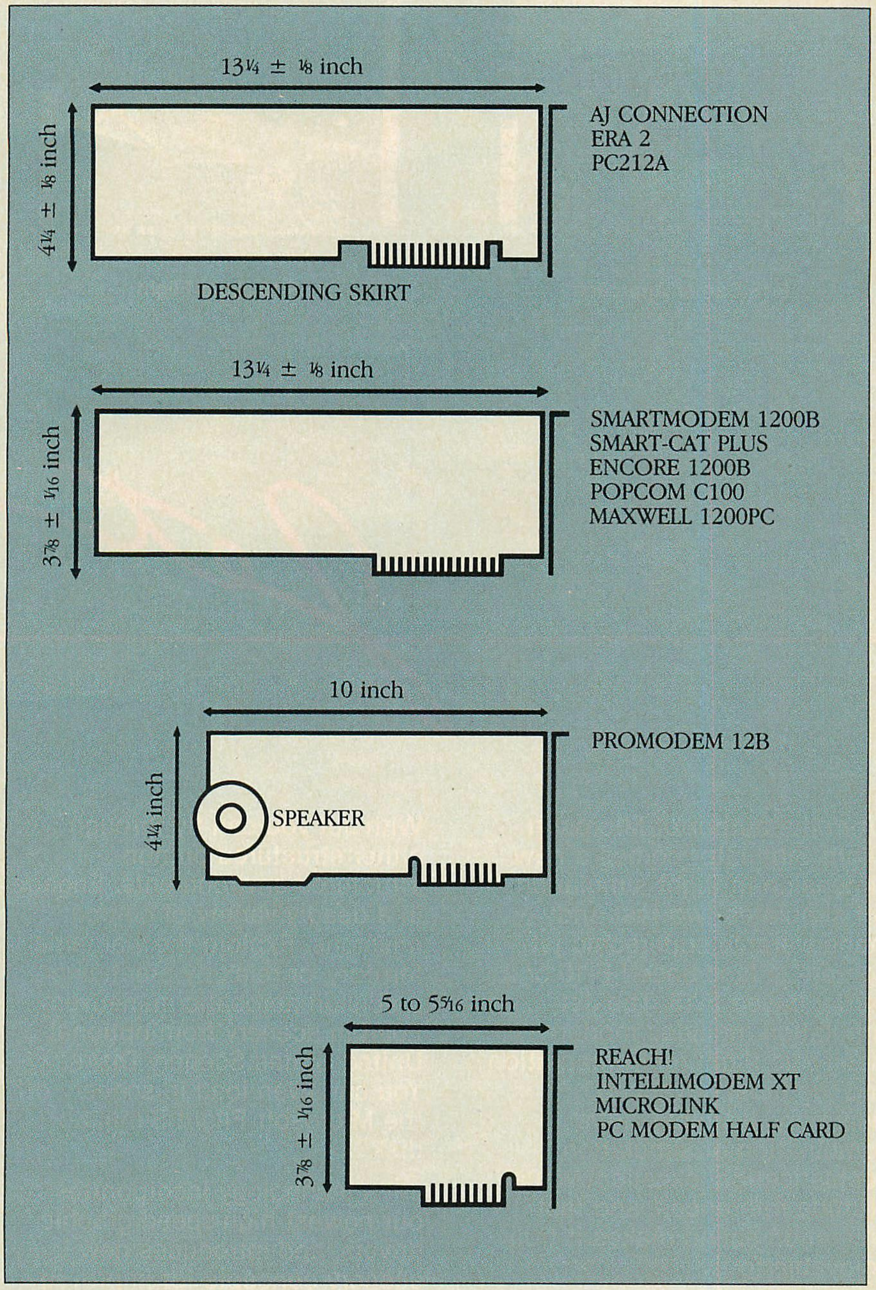
The recessed modular telephone jacks on eight of the boards are quite annoying. Removing a modular plug when the locking tab is hidden under the surface of the board's mounting bracket is very difficult. Anderson Jacobson, Bizcomp, Hayes, OmniTel, and PROMETHEUS are to be congratulated for flush mounting their telephone jacks for easy access. The Hayes jacks actually protrude slightly, making for even easier access to the locking tabs.

To gain some product differentiation, both RIXON and Anderson Jacobson add a second serial interface to their boards, which brings out a familiar 25-pin connector on the rear mounting bracket. OmniTel offers an optional auxiliary serial port for its modem board, also. The ports can be freely assigned to available address spaces and interrupt levels using jumpers and switch settings. The auxiliary port circuits can be disabled if the host system has a serial port on another board.

All of the modems provide the basic and extended forms of *result codes* in response to modem commands. (For more information on the Hayes command set, see the sidebar.) AST, Novation, Prentice, and Racal-Vadic provide special result codes for remote ring, busy, hang-up, and no answer. Prentice's POPCOM can even report that a connection sounds like it is carrying voice traffic instead of data, and it can report a bad line, lack of expected dial tone, and a local phone off-hook condition. Such result codes should be ignored by programs that do not expect them, or they may be dumped arbitrarily on the screen. Programs that can interpret such codes, however, can do far more sophisticated error handling and status reporting than is possible with the Hayes result code set.

Table 3 compares the software that is bundled with each of the products (only the POPCOM C100 offers none). Microstuf's popular Crosstalk XVI, version 3.5, is included in five of the products: AJ Connection, AST Reach!, Encore 1200B, PC212A, and PC Modem Half Card. Among Crosstalk's significant features are the RUN and EDIT commands, which run DOS and stand-alone programs from inside Crosstalk; the file-transfer protocol XMODEM; and VT100 terminal emulation.

FIGURE 1: *Modem Profiles*

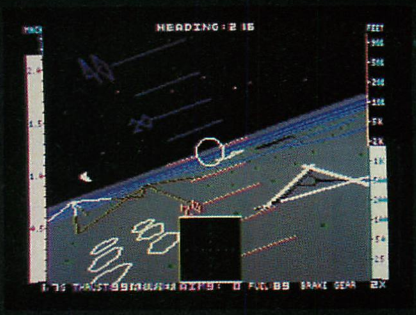


Four of the 13 modems are half-sized cards, permitting their use in the PC, XT, Portable PC, and AT because they can fit in the narrow, short slots. Unfortunately, those boards with a descending skirt can only be seated in slots 1 and 7 on the AT. Those that are thicker than three-fourths of an inch will not fit comfortably in the narrow board-to-board spacing of the XT, Portable PC, and AT.

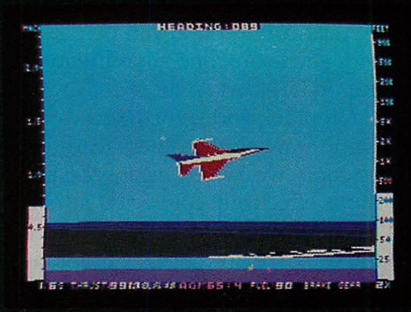
For file-transfer operations, MITE/MS, supplied with Smart-Cat PLUS and ProModem 1200B, offers more choices than the competition. However, it offers no terminal emulation (except dumb) capabilities. The Hayes Smartcom II package accompanies the Smartmodem 1200B. It includes XMODEM protocol transfers and a VT100-series emulation among its capabilities.

The Era 2 package uses the Microcom Networking Protocol (MNP) that is gaining favor in many quarters for use in electronic message systems and other applications in which reliable, unat-

F-18 Carrier Approach



F-16 Night Flight



F-16 from Control Tower

Jet

From the author of Microsoft Flight Simulator comes a new dimension in flight simulation. Jet simulates two fast and maneuverable supersonic jet fighters, a land-based F-16 or a carrier-based F-18. The Jet simulation includes modern electronic flight instrumentation and the most advanced weaponry available. Jet's combination of simulation sophistication, excellent visual attitude references, and fast animation speed makes it truly enjoyable to fly.

Excellent aircraft control plus ballistic thrust gives you the kind of aerobatic maneuver-ability only a modern jet fighter can provide. The attitude indicator is easy to read no matter what your orientation. This indicator, combined with a full-screen out-the-window view, helps you get the most out of Jet's excellent flight controls. An important consideration when traveling at speeds in excess of 1300 MPH.

With Jet you can fly through either structured on non-structured environments. Engage in a deadly variety of combat missions. Explore the wargame territory, or relax by practicing precision aerobatic maneuvers. Load in scenery from optional United States Scenery Disks. You can even load in scenery off the Microsoft Flight Simulator disk.

Jet's technology incorporates our newest high-speed graphic drivers, and introduces new scenery concepts for improved viewing on non-color and RGB monitors.

Jet requires an IBM™ PC, PC jr, PC AT, Compaq™, or Tandy™ 1000-series computer, with 128K minimum memory and the Color/Graphics Monitor Adapter, Hercules™ Graphics Card, IBM Enhanced Graphics Adapter, or Sublogic High-Speed Graphics Card.

See Your Dealer . . .
or write or call for more information. For direct orders please enclose $49.95 plus $2.00 for shipping and specify UPS or first class mail delivery. Visa, MasterCard, American Express, and Diners Club cards accepted.

SCENERY DISKS FOR MICROSOFT FLIGHT SIMULATOR AND JET ARE NOW AVAILABLE.

IBM is a trademark of International Business Machines Corp.
Compaq is a trademark of Compaq Computer Corp.
Tandy is a trademark of Tandy Corp.
Hercules is a trademark of Hercules Computer Technology.

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

Order Line: (800) 637-4983
(except in Illinois, Alaska, and Hawaii)

CIRCLE NO. 145 ON READER SERVICE CARD

tended file transfers are required. IntelliModem's software is Intellisoft; Racal-Vadic supplies GEORGE; and U.S. Robotics packages Microlink with TELPAC. (For more information about many of these communications programs, refer to the following *PC Tech Journal* articles: "Crosstalking," Augie Hansen, July 1984, p. 90; "ACS or PCM: The Better to Communicate With," Augie Hansen, August 1984, p. 155; and "Screenspeak, " Augie Hansen, November 1984, p. 151.)

The registers and commands used by Hayes form the basis of comparison for these internal modems for PCs. Table 4 summarizes the availability of registers in each of the tested products, and table 5 shows the commands that each accepts. Together, these tables give a useful picture of the degree of Hayes compatibility for each modem.

Not all Hayes registers need to be duplicated for a modem to work successfully with software designed for the Smartmodem. The tack taken by most manufacturers is to ignore impossible requests and return the reassuring but deceptive OK result code. An example of this behavior is the dial-tone wait period set by register S6. Values in the range of 2 to 255 seconds are allowed, and 2 seconds is the usual default. Some modem makers simply set this value to a few seconds, allowing no changes, and by so doing satisfy the needs of most modem users.

A similar situation occurs with commands. The Cn command that can be used to control the ON/OFF condition of the modem transmitter is not implemented on Maxwell 1200PC, Microlink, PC212A, and POPCOM.

INSTALLATION AND SET-UP

The job of installation and set-up can be time-consuming and troublesome, depending on the modem. A few of the modem manufacturers (Microcom, Hayes, AST) make the job of finding needed information about port and interrupt assignments in the manuals quite easy. A few others (Ven-Tel, Anderson Jacobson) turn what should be a simple job into an adventure.

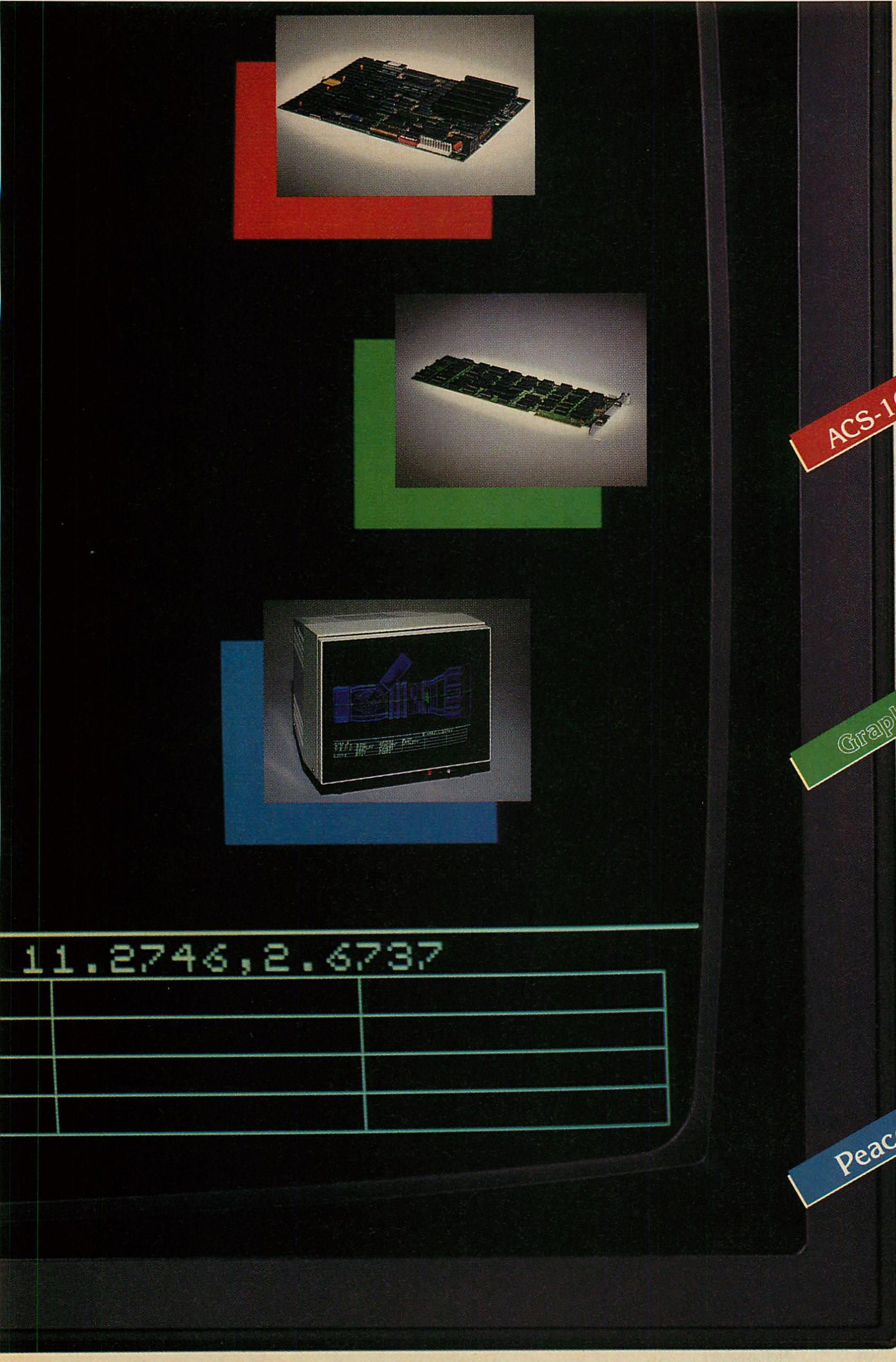
Just trying to find out which jack is for the line connection and which for the optional telephone for nearly half of these modems can be frustrating. Labeling the jacks would be a simple matter. The manufacturers who leave the labels off are usually the ones who bury the instructions in a nonindexed manual, compounding the error. (Table 2 identifies these vendors with a dash in the row that is marked "line connections: labeled," meaning they are not).

TABLE 2: *Hardware Features Supported*

	AJ CONNECTION	ENCORE 1200B	ERA 2	INTELLIMODEM XT	MAXWELL 1200PC	MICROLINK	PC212A	PC MODEM	POPCOM C100	PROMODEM 1200B	REACH!	SMART-CAT PLUS	SMARTMODEM
TRANSMISSION RATE													
0-300 bps	o	o	o	o	o	o	o	o	o	o	o	o	o
1200 bps	o	o	o	o	o	o	o	o	o	o	o	o	o
CHANNEL CONTROL													
Half-duplex	o	o	o	o	o	o	o	o	o	o	o	o	o
Full-duplex	o	o	o	o	o	o	o	o	o	o	o	o	o
DIALING													
Pulse	o	o	o	o	o	o	o	o	o	o	o	o	o
Touch-tone	o	o	o	o	o	o	o	o	o	o	o	o	o
Mixed	o	o	o	o	o	o	o	o	o	o	o	o	o
MODE													
Originate	o	o	o	o	o	o	o	o	o	o	o	o	o
Auto-answer	o	o	o	o	o	o	o	o	o	o	o	o	o
Reverse	o	o	o	o	–	–	o	o	o	o	o	o	o
SELF-TEST	o	o	–	o	o	–	o	o	o	–	o	o	o
SPEAKER													
On-board	o	o	o	o	–	o	–	o	o	o	o	–[a]	o
Volume control	–[b]	–[b]	–	o	–	o	–	–	o	o	o	o	–
On/off control	o	o	o	o	–	o	–	o	o	o	o	o	o
LINE CONNECTIONS													
Jack mounting[c]	F	F	R	R	R	R	R	R	R	F	R	R	P
Labeled	o	o	o	–	o	–	–	–	–	–	o	o	o
Single (RJ11)	o	o	o	o	o	o	o	o	o	o	o	o	o
Multiline	–	–	–	o	–	–	–	–	o	o	–	o	o
SECOND RS-232 PORT	–[d]	–[d]	–	–	–	–	o	–	–	–	–	–	–
CALL PROGRESS REPORTING													
Dialtone	–	–	–	–	–[e]	–	–	–	o	–	o	o	–
No tone	–	–	–	–	–[e]	–	–	–	o	–	–	–	–
Phone off-hook	–	–	–	–	–	–	–	–	o	–	–	–	–
Number busy	–	–	–	–	–[e]	–	–	–	o	–	o	o	–
Number ring	–	–	–	–	–[e]	–	–	–	o	–	o	o	–
No answer	–	–	–	–	–[e]	–	–	–	o	–	o	–	–
Voice	–	–	–	–	–	–	–	–	o	–	–	–	–
Line bad	–	–	–	–	–	–	–	–	o	–		–	–

[a] *Uses PC system unit speaker via a supplied cable.*
[b] *Speaker volume controlled by software.*
[c] *Jack mounting position: R = recessed, F = flush, P = protruding.*
[d] *Second serial port is an extra cost option.*
[e] *Provided in proprietary mode but not in Hayes-compatibility mode.*

All of the modems can transmit data at 110, 300, and 1200 bits per second (bps) in either full- or half-duplex operation. All can use pulse or tone dialing or a combination of the two. Most have an on-board speaker. All support a single-line connection, and five of them also support multiline connections.



PROUD.

To build a workstation good enough to put your name on, use the modular building blocks we put our name on. ACS, a company dedicated to the support of OEM's in pursuit of excellence.

ACS-1000

THE POWER . . . a single board SuperComputer that is **100% IBM XT compatible** from its mounting holes to its memory mapping, but with more speed: 4.7 and **8 MHz.** Floppy controllers and network communications are built right onto the motherboard along with a SASI hard disk interface and as much as **1 Meg** of memory.

GraphAx

THE GLORY . . . a graphics card (one card) with more than **20,000,000 pixels** of video memory powerfully arrayed in 5 planes **(32 Colors).** Driven by an on board graphics processor, this **2K x 2K** color card is compatible with most of the popular CAD/CAM packages . . . like AUTOCAD, PC Draft, GKS, and Tektronics Emulators . . . packages that run up to 5 times faster and virtually eliminate redraw by **zooming and panning** in memory instead of constantly recalculating.

Peacock

AN IMPRESSIVE DISPLAY
The Peacock high resolution monitor will put all this technology up front where it counts. With a horizontal scanning frequency of **33.5 KHz** and a bandwidth of **40 MHz,** this RGB monitor gives you a **1K x 1K** window on the world with profitable OEM pricing.

If you're going to build a workstation, you might as well build the best. The people at ACS are ready to help. For more information, call or write:

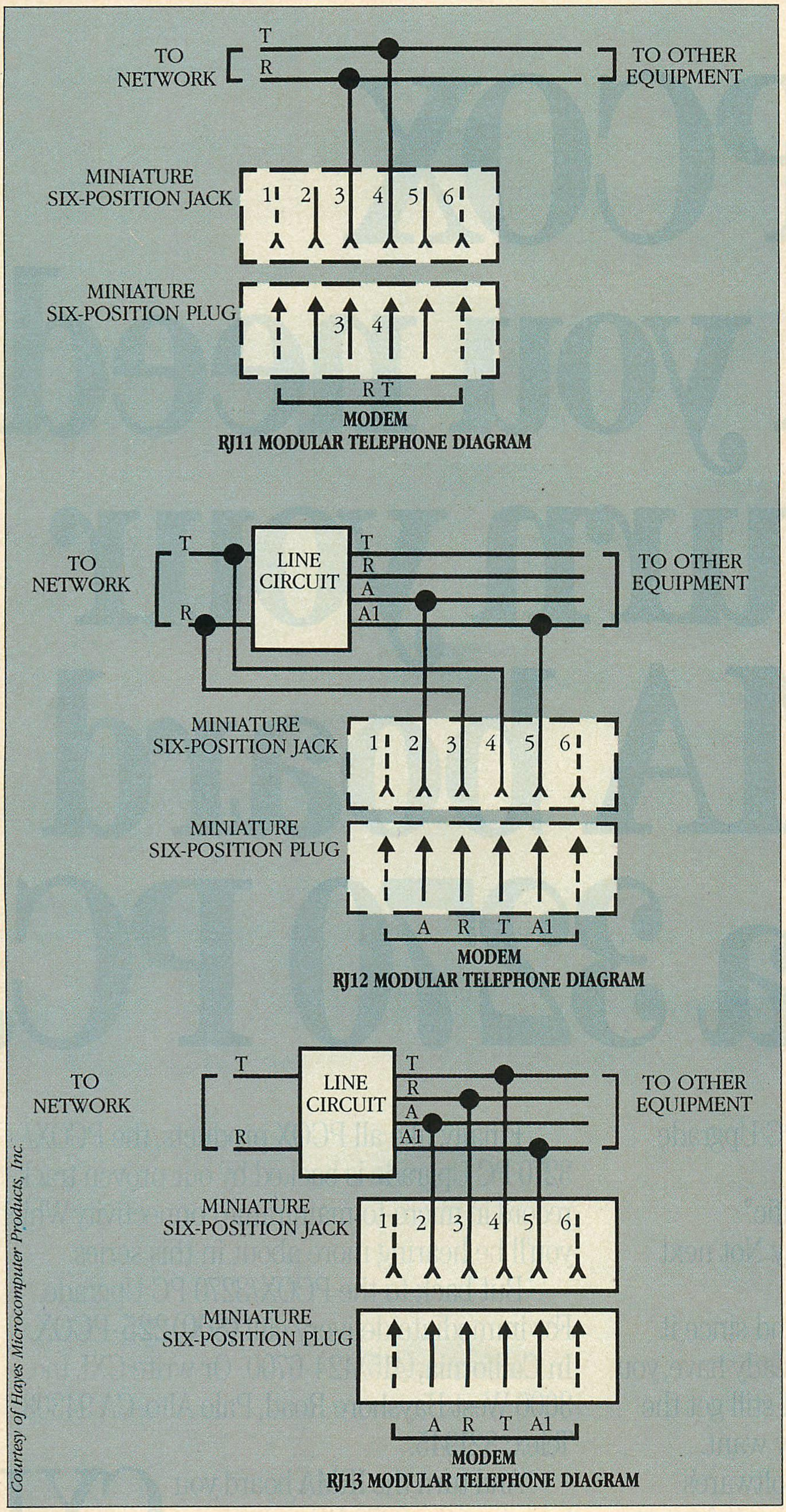
ACS International
2105 Luna Rd. Suite 330
Carrollton, Texas 75006
214-247-5151
TLX 709748 ACS UD

In Canada:
Soltech Industries
9274 194th St.
Surrey, B.C. V2T4W2
604-888-2606

CIRCLE NO. 102 ON READER SERVICE CARD

ACS

FIGURE 2: *Single- and Multiline Connections*



All of the modems work with the RJ11 connection, a two-wire feed that services a standard, nonelectronic telephone. Only five of them work with an RJ12 or RJ13 connector. These modems use the A-lead control of multiline feeds, a four-wire arrangement that permits signaling for the control of busy lamps when at least one of the connected telephone sets is off-hook.

The packaging and presentation of the manuals delivered with the modems are, for the most part, excellent. Quality of content and organization vary, but are generally good. The few exceptions include Ven-Tel, whose documentation is only marginally acceptable, with numerous omissions and some misleading information. Both the U.S. Robotics and Anderson Jacobson manuals are densely written and difficult to follow. The OmniTel manual lacks information about commands and registers that would enable a programmer to find out what is implemented and what is not.

An indexed manual is essential for topics as complex as communications and modem technology. Modems for which the documentation lacked a detailed table of contents or a useful index lost some of the luster they had acquired in other ways. Both Hayes and AST are guilty of not putting indexes in their hardware manuals, which are in other respects very well done.

THE MODEM SCORECARD

What follows is a capsule summary of the findings for each of the tested internal modem packages. As in the tables, the order of presentation is alphabetical by product name.

During tests over local and long-distance connections, none of the results could be duplicated. All the modems produced erroneous data at times. None was really bad and none was perfect, although the Racal-Vadic Maxwell Modem 1200PC came close. Nearly all of the modems use the same basic techniques—Frequency Shift Keying (FSK) for 0-300 bps, four-level Phase Shift Keying (PSK) for 1200 bps—and respond similarly to line noise and signal interruptions. The exception was Racal-Vadic's Maxwell 1200PC, which uses a coherent detection scheme for improved receiving accuracy.

The worst-case connection for the tests was a line with recurring clicks, apparently caused by central office signaling. The clicks are disturbing to humans—moreso to modems; all of the products delivered junk to the screen when a click occurred, and all hung up during protocol file transfers.

AJ Connection. The Anderson Jacobson card is full-length with a descending skirt, a built-in speaker (slightly raspy), and an auxiliary serial port. Configuration switches can be accessed at the top edge of the board without removing it. Either port may be assigned to COM1 through COM4. The COM3 and COM4 assignments are available on some compatible computers (Columbia 1600, for

example) and may be established on the PC using IRQ2 and IRQ5 in addition to the IRQ3 and IRQ4 used by IBM. The full Hayes command set is provided; only bit-mapped register S17 is omitted.

This modem features a software volume control. It is directly controlled using a new command, Ln, where *n* ranges from 0 to 7. Crosstalk XVI, version 3.50-AJ, is a modified version of that program that handles the volume setting in a set-up file and facilitates use of COM3 and COM4 port allocations.

The hardware manual is densely written and its production is marginal at best—a contrast next to the Crosstalk documentation. A handy flip-easel binder and slipcase are provided.

Encore 1200B. OmniTel produces two versions of this full-sized board: a modem-only configuration and one that comes with a second serial port. The modem-only board was tested for this article. The jacks are flush-mounted and clearly marked; the serial connection is broken when data transmission begins. The built-in speaker's volume can be adjusted via software.

This is another modem that permits access to COM3 and COM4 ports by using interrupt levels IRQ2 and IRQ5 and additional address spaces at 3E8H-3EFH and 2E8H-2EFH. Crosstalk XVI, provided with Encore, allows selection of any one of the four COM ports.

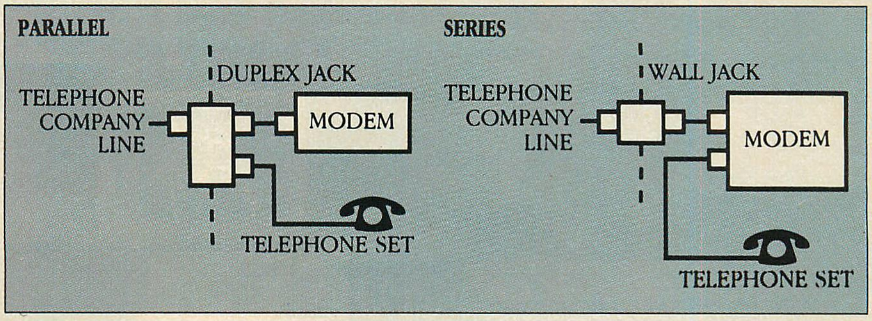
The documentation provided by OmniTel reproduces the Crosstalk XVI documentation and fully describes the register and command sets, which are completely Hayes-compatible. The typeset manual is comb-bound.

Era 2. The Era 2 card from Microcom is full-size and fits only in slot 1 of an AT because of a descending skirt. Two jacks are recessed but well marked, and the modem breaks the connection to a series-connected telephone. The board is seven-eighths inches wide and requires two slots in machines with narrow board-to-board spacing. Circuitry on the board shows the existence of a second serial port, which is not described in the documentation. No components are installed for it.

This modem has both a Hayes-compatible and a proprietary command set. Only the Fn command, duplex selection, is omitted. Registers S11 for Touch-Tone timing and S16 for self-test are also missing. An Era 2 software package features the Microcom Networking Protocol and VT100-series and IBM 3101 terminal emulations.

Documentation is complete and well-produced in most respects. It has a detailed table of contents and a sparse but useful index. The product has the longest duration limited warranty of any modem tested—four years.

FIGURE 3: *Parallel and Series Connections*



A telephone may be installed in tandem with the modem in two ways. A duplex jack may be used to place the phone in parallel with the modem, as shown at the left, or the telephone may be placed in series with the modem by plugging it into the phone jack provided on all of the modems, as shown at the right.

TABLE 3: *Communications Program Features Supported*

	CROSSTALK XVI	ERA 2	GEORGE	INTELLISOFT	MITE/MS	TELPAC	SMARTCOM II
FILE-TRANSFER PROTOCOL							
CLINK	–	–	–	–	o	–	–
Crosstalk	o	–	–	–	o	–	–
Hayes	–	–	–	–	o	–	o
XMODEM	o	–	o	o	o	o	o
XMODEM/B (multifile)	–	–	–	–	o	–	–
MNP	–	o	–	–	–	–	–
MITE	–	–	–	–	o	–	–
Start/stop	o	o	o	o	o	o	o
Send lines	o	o	o	o	o	o	o
TERMINAL EMULATION							
VT100/52	o	o	–	–	–	–	o
TeleVideo 910/20	o	–	–	–	–	–	–[a]
ADM3a	o	–	–	–	–	–	–
3101	o	o	–	–	–	–	–
TI914	o	–	–	–	–	–	–

[a] *A subset of capabilities is emulated.*

By far the most popular communications package bundled with the modems is Crosstalk XVI, included with AJ Connection, Reach!, Encore 1200B, PC212A, and PC Modem Half Card. Other packages are Microcom Era 2 with Era 2; GEORGE with Maxwell 1200PC; Intellisoft with IntelliModem XT; MITE/MS with Smart-Cat PLUS and ProModem 1200B; TELPAC with Microlink; and Smartcom II with Smartmodem 1200B. Only one modem, the POPCOM C100, does not bundle any software.

IntelliModem XT. Bizcomp's modem features full Hayes register and command sets in a good-quality, half-sized board. It has two unmarked, recessed jacks and no phone disconnect feature. A small, tinny-sounding speaker is provided for audio monitoring. A volume control is accessible on the card adapter bracket.

Intellisoft is Bizcomp's software package for the IntelliModem. It has basic communications and XMODEM file transfers. Documentation is presented in a spiral-bound book, and the manuals are fully indexed.

Maxwell 1200PC. Racal-Vadic has produced a mute modem, with no speaker and no provision to use the one in the host computer. The board is full-length with a partial descending skirt, used

Networking Raised to a Greater Power

Advanced Technology. With it, IBM tripled the speed of the PC and increased its memory capacity five-fold. Nowhere is this increase in computing power more important than in networking situations. If the AT's technological advances have prompted you to look into a multi-user network, you owe it to yourself to take a closer look at MultiLink Advanced™ . . . a unique multi-tasking, multi-user networking system that runs programs under PC-DOS 3.0.

Eight Workstations for the Price of an AT. MultiLink Advanced™ represents the next generation in networking systems for IBM microcomputers. The system enables terminals, connected to a single AT, to emulate IBM-PC's having up to 480K of RAM (The PC-Shadow™ terminal, shown above, even has a PC look-alike, as well as work-alike, keyboard and display).

This means that instead of spending $3,000 per workstation for a PC with a Kilobuck "Network Interface Board," you can use inexpensive terminals . . . eight of which cost less than an IBM AT. Even if you need only one workstation connected to your AT, you'll realize significant savings.

MultiLink Advanced™ . . . Instant Access to All of Your Resources. Central to most multi-user situations is the need to coordinate a variety of printers. With what's been described by *PC-Tech Journal* as ". . . by far, the best print spooler for the IBM PC," MultiLink Advanced™ gives users the option to print either at their workstations, or at a central location. In addition, programs and files can be shared by multiple users locally or through use of a modem. Just think of it . . . having remote access to an AT with a lightweight terminal/modem.

Although designed to take advantage of the AT, MultiLink Advanced™ runs on all versions of PC-DOS, except 1.0, and certain implementations of MS-DOS. A wide range of leading programs are supported which include WordStar, dBASE III, Multimate, and Lotus 1-2-3.

Get the Advanced Story Today. Call The Software Link Today for complete details and the dealer nearest you. MultiLink Advanced™ is immediately available at the suggested retail price of $495 and comes with a money-back guarantee. VISA, MC, AMEX accepted.

MultiLink™
ADVANCED

THE SOFTWARE LINK, INC.

8601 Dunwoody Place, Suite 336, Atlanta, GA 30338 Telex 4996147 SWLINK
CALL: 404/998-0700

Dealer Inquiries Invited

MultiLink Advanced™ is a trademark of The Software Link, Inc.

IBM, PC, AT, & PC-DOS are trademarks of IBM Corp. MS-DOS, WordStar, dBASE III, Lotus 1-2-3, and Multimate are trademarks of Microsoft Corp., MicroPro, Ashton-Tate, Lotus Development Corp., & Multimate International, respectively.

CIRCLE NO. 143 ON READER SERVICE CARD

only to hold a serial number tag. The skirt, however, does not interfere with its installation in some AT slots, because it wedges in between memory chips on the main board. It does cause the board to be tilted in its socket in some slot positions in which the skirt rests on top of the memory chips.

Racal-Vadic provides its own communications software, called GEORGE. While hardly an exceptional program, it does handle basic communications needs and XMODEM file transfers. No terminal emulations are provided. GEORGE is needed because other programs, such as Crosstalk XVI, do not use the special call-progress codes that Maxwell issues in its proprietary mode (initiated by a Ctrl-E command). The serial telephone connection is not cut off when the modem switches to data operation. The two slightly recessed jacks are clearly marked.

The Maxwell modem brings industrial strength communications to the PC realm by incorporating the same coherent detection scheme that is the mainstay of Racal-Vadic's line of industrial modems. The technique is based on a crystal-controlled optimal reference signal that permits recovery of the desired signal from very noisy channels.

The documentation is well-organized and illustrated. The hardware portion lacks an index.

Microlink. U.S. Robotics offers a half-sized board with two recessed, unmarked jacks and an on-board speaker with an externally accessible volume control. Very limited support for Hayes registers (S0, answer on ring number; S2, escape code character; and S7, wait for carrier) is provided. Attempts to alter missing registers are ignored and return an OK result code. The modem command set lacks **C**, transmitter carrier control; ;, command state return; **H**, on/off hook; **I**, identification; and **O**, online return. The modem does not work with multiline connections and has no provision for reverse connection to an originate-only modem.

Microlink is packaged with a proprietary communications package called TELPAC that has little to recommend it except the XMODEM file-transfer protocol. TELPAC has no terminal emulations, and its operations are rather unsophisticated. It cannot be used reliably with any other modems.

The documentation for this product is awkward in its organization, lacks indexes and dividers, and has marginal production quality.

PC212A. The PC212A from RIXON is built on a full-length card that has a partially

TABLE 4: *Modem Registers Supported*

REGISTER	AJ CONNECTION	ENCORE 1200B	ERA 2	INTELLIMODEM XT	MAXWELL 1200PC	MICROLINK	PC212A	PC MODEM	POPCOM C100[a]	PROMODEM 1200B	REACH!	SMART-CAT PLUS	SMARTMODEM
S0	o	o	o	o	–	o	o	o	o	o	o	o	o
S1	o	o	o	o	o	–	–	o	o	o	–[b]	o	o
S2	o	o	o	o	o	o	–	o	o	o	o	o	o
S3	o	o	o	o	o	–	–	o	o	o	o	o	o
S4	o	o	o	o	o	–	–	o	o	o	o	o	o
S5	o	o	o	o	o	–	–	o	o	o	o	o	o
S6	o	o	o	o	o	–	–[d]	o	o	o	o	o	o
S7	o	o	o	o	o	o	–[d]	o	o	o	o	o	o
S8	o	o	o	o	o	–	–[d]	o	o	o	o	o	o
S9	o	o	o	o	–	–	–[d]	o	–	o	o	o	o
S10	o	o	o	o	o	–	–[d]	o	o	o	o	o	o
S11	o	o	–	o	o	–	–[d]	o	–	o	o	–	o
S12	o	o	o	o	o	–	–[d]	o	o	o	o	o	o
S13	o	o	o	o	–	–	–	o	–	o	o	o	o
S14	o	o	o	o	–	–	–	o	–	o	o	o	o
S15	o	o	o	o	–	–	–	o	–	o	o	o	o
S16	o	o	–	o	–	–	–	o	o	–[c]	o	o	o
S17	–	o	o	o	–	–	–	–	–	–	–	o	o

[a] *POPCOM has a register S18 that is used to control EIA signals.*
[b] *This register is fixed at 000 and is read-only.*
[c] *Self-test function is provided by an @Xn command.*
[d] *These registers are read-only and have preset constant values.*

Whether the internal modem supports the Hayes set of registers is a good measure of its programming compatibility. See the accompanying sidebar on the Hayes standard for an explanation of the register usage.

descending skirt. The version tested has an auxiliary serial port but no speaker on the card and no PC audio connection. The modem supports only single-line connections. The two unmarked jacks are slightly recessed and the series telephone connection is not broken when the modem is in data mode.

The package includes Crosstalk XVI. The documentation is in a standard Crosstalk binder to which a hardware manual has been added. The hardware manual has a useful glossary and table of contents, but no index.

The only Hayes register that is implemented is S0, answer on ring number. All the others are either missing or hard-coded with default values that cannot be changed. RIXON provides an on-board memory system with battery back-up and a Z80A microprocessor that provides auto-dialing and other communications facilities without an external program. Support for the Hayes command set is complete except for the Cn command, transmitter carrier control.

PC Modem Half Card. Ven-Tel is responsible for pushing every manufacturer in the direction of full-function, half-sized cards with its entry in the internal modem market. PC Modem has nearly complete register and command sets. Only bit-mapped register S17 and the I (identification) command are missing. The small speaker provided with PC Modem has perhaps the poorest sound quality of those tested. The two recessed jacks are not marked. The telephone connection is a parallel arrangement.

The PC Modem package includes Crosstalk XVI and has a manual of marginal quality. Finding accurate information about the line and telephone jacks in the manual is nearly impossible.

POPCOM C100. This is the only unit tested that has no companion software. Oddly, its price does not reflect that omission. The board occupies a full slot, has two recessed, unmarked connectors, and contains a clicky-sounding speaker.

Some Hayes registers and commands are not supported. Additional

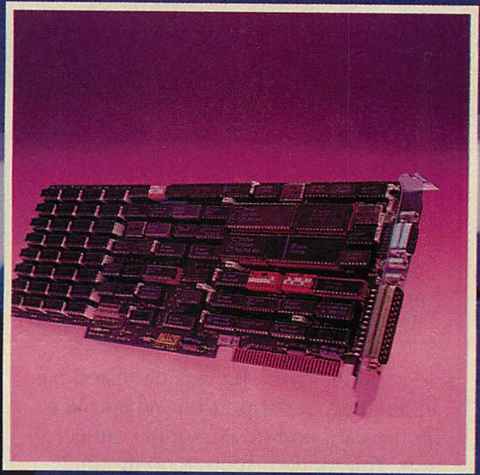
Alloy makes your PC multi-user.

Introducing Alloy's PC-SLAVE/16. The multi-user PC strategy that doesn't require additional PCs.

PC-SLAVE/16 transforms your IBM PC into a multi-user, multi-processor system with shared data access and communication between users. All you need is a dumb terminal and a PC-SLAVE/16 board for each user.

And with Alloy's PC-X BUS, you can add up to 31 users. So whether you're adding your secretary or 31 colleagues, Alloy's PC-SLAVE/16 can do it at a much lower cost and with higher performance.

For any system requiring more than one PC, PC-SLAVE/16 is more powerful and less expensive. Each board comes complete with its own 8 MHz, 8088 micro processor and dedicated memory. PC-SLAVE/16 eliminates the need to add costly PCs. And it is ten times faster than LANs (Local Area Networks) at half the cost. Best of all, it's available right now. See PC-SLAVE/16 in action at your local dealer. Or call Alloy today at (617) 875-6100.



ALLOY

ALLOY
Computer Products, Inc.

Alloy Computer Products, Inc., 100 Pennsylvania Avenue, Framingham, Massachusetts 01701. (617) 875-6100, TWX: 710-346-0394
In Europe: Alloy Computer Products (Europe) Ltd., Cirencester, Gloucestershire, England. Tel: 0285-68709, Tlx: 43340

CIRCLE NO. 112 ON READER SERVICE CARD

commands and one extra bit-mapped register are provided partially to compensate for this. The modem ran well with Crosstalk XVI, version 3.5.

The documentation is good, but lacks an index. Useful information about multiline connections is provided, including a note to the telephone installer describing some recommended arrangements.

ProModem 1200B ComCard. The PROMETHEUS modem is roughly a two-thirds card, measuring 10 inches deep. It is about seven-eighths of an inch wide, so it requires two adjacent slots in an XT, Portable PC, and AT. The two flush-mounted jacks are not marked. They are held in place on the circuit board by a nylon strap that makes the board thickness at that point closer to one inch. The telephone jack is disconnected when the modem goes off-hook.

This modem has no self-test mode. It has a set of extended result codes that include DISCONNECT, INVALID COMMAND, NO DIAL TONE, COMMAND > 40 CHARS, DIAL COMPLETE, RINGING, and BUSY. The modem can be configured for COM3 if necessary.

The ProModem is packaged with MITE/MS software. It contains good, indexed documentation.

Reach!. AST has produced a high-quality, half-sized board that, operationally, is almost identical to the Hayes Smartmodem. Reach! has two recessed jacks, well-marked, and breaks the phone connection when the modem is activated. A volume control and some configuration switches are accessible on the card-mounting bracket. Register S1 always returns 000; it does not count the number of rings. A switch setting tells the modem to answer either on the first ring or not at all. A nonzero value in register S0, the number of rings on which the modem answers, will override the switch setting. Bit-mapped register S17 is not implemented.

Reach! has a menu-driven test program that aids users in determining the cause of problems. Both local originate and answer tests and a remote digital loopback test are provided, in addition to dial tone, busy, and carrier tests. The program requires a BASIC interpreter.

A cable for an RJ11 connection is supplied with a toroidal RF choke (doughnut-shaped ferrite) to prevent stray radio frequency energy from getting into the modem. This is important to users whose systems are immersed in strong RF fields, for example, from nearby radio stations.

The documentation is well written and nicely produced, including tabbed dividers and good use of color and illustrations. The hardware portion of the manual contains no index. Crosstalk XVI is provided as operating software.

TABLE 5: *Modem Commands Supported*

COMMAND	AJ CONNECTION[a]	ENCORE 1200B	ERA 2	INTELLIMODEM XT	MAXWELL 1200PC	MICROLINK	PC212A	PC MODEM	POPCOM C100	PROMODEM 1200B[b]	REACH!	SMART-CAT PLUS	SMARTMODEM
AT	o	o	o	o	o	o	o	o	o	o	o	o	o
+++	o	o	o	o	o	o	o	o	o	o	o	o	o
A	o	o	o	o	o	o	o	o	o	o	o	o	o
A/	o	o	o	o	o	o	o	o	o	o	o	o	o
Cn	o	o	o	o	—	—	—	o	—	o	o	o	o
,	o	o	o	o	o	o	o	o	o	o	o	o	o
D	o	o	o	o	o	o	o	o	o	o	o	o	o
En	o	o	o	o	o	o	o	o	o	o	o	o	o
Fn	o	o	—	o	—	o	o	o	—	o	o	o	o
Hn	o	o	o	o	o	o	o	o	—[c]	o	o	o	o
In	o	o	o	o	—	—	o	—	—[d]	o	o	o	o
Mn	o	o	o	o	—	o	o	o	o	o	o	o	o
O	o	o	o	o	o	—	o	o	o	o	o	o	o
P	o	o	o	o	—[e]	o	o	o	o	o	o	o	o
Qn	o	o	o	o	o	o	o	o	o	o	o	o	o
R	o	o	o	o	o	o	o	o	o	o	o	o	o
Sr?	o	o	o	o	o	o	o	o	o	o	o	o	o
Sr=	o	o	o	o	o	o	o	o	o	o	o	o	o
;	o	o	o	o	o	—	o	o	o	o	o	o	o
T	o	o	o	o	—[e]	o	o	o	o	o	o	o	o
Vn	o	o	o	o	o	o	o	o	o	o	o	o	o
Xn	o	o	o	o	—	o	o	o	o	o	o	o	o
Z	o	o	o	o	o	o	o	o	o	o	o	o	o

[a] *The AJ Connection adds a G command for remote digital loopback testing and an L command to control speaker volume.*
[b] *PROMETHEUS has added several @ commands; @H-help, @T-time, @T=-set time, @Xn-modem loopback test.*
[c] *H2 not supported.*
[d] *The command meaning has been modified by Prentice.*
[e] *P and T may be used only within a dialing command string.*

The availability of registers and commands used by Hayes form the basis of comparison for these internal modems for the PC family of computers. See the sidebar on the Hayes standard for an explanation of each command.

Smart-Cat PLUS. Novation's modem occupies a full slot. It has two recessed jacks and a volume control, but no speaker; instead, it uses the PC's speaker via a jumper cable. The serial telephone connection is broken whenever data communications are begun.

MITE version 2.81, the communications program provided with the Smart-Cat PLUS, is not copy-protected, but it insists on checking for a diskette—any diskette—in drive A:. If one is not found, such as when a user is running entirely from a hard disk, MITE will not start. The program is long on file-transfer protocols (six) and short on terminal emulations (none except dumb tty). It has a menu-driven user interface. The modem and MITE support the use of a third port, COM3.

The hardware and software manuals have no indexes, but otherwise are of good quality. Schematic diagrams are shown for the modem, and the hardware manual also contains an excellent discussion of multiline connections.

Smartmodem 1200B. Hayes Smartmodem 1200B is the de facto standard and thus can be used as a basis for comparison. The version 2.0 unit being sold now is narrow enough to fit in the XT, AT, and Portable PC.

Smartcom II is provided to manage communications activities. It works well with Smartmodem but may not work with other hardware that is not truly Hayes-compatible.

RECOMMENDATIONS

The convenience offered by internal modems is enticing. The best internal modem would be either a half-sized card that is truly compatible with the Hayes standard and of comparable quality or a full-sized card with an extra feature, such as an auxiliary serial port.

The U.S. Robotics and Ven-Tel offerings deserve a thumbs-down rating based on the marginal quality of the documentation. Conversely, AST Reach! and Hayes Smartmodem 1200B have good documentation, indicative of a dedication to excellence. This quality shows in nearly every aspect of the hardware and software and, most importantly, in customer support.

If AST would add the multiline feature, put an index in the hardware portion of the otherwise excellent manual, and flush-mount the telephone and line jacks, it would be an excellent product. Bizcomp is also worthy of serious consideration. The Intellisoft program is no match for Crosstalk XVI, but the IntelliModem XT hardware is comparable to that of Reach!, and it has the multiline feature and a decent index. Alas, it too has recessed jacks.

The full-sized boards should not be discounted. Many PC owners do not have short slots to influence their buying decisions, and all of the boards performed well enough to warrant consideration. The products packaged with Crosstalk XVI have an edge because of the overall versatility of such combinations. For highly reliable communications in extremely noisy environments, Racal-Vadic's Maxwell 1200PC is the leading contender.

Augie Hansen owns Omniware, a software development and training firm specializing in UNIX and MS-DOS systems and applications. He has written several articles on communications programs for PC Tech Journal.

THE HAYES STANDARD

Although Hayes Microcomputer Products Inc. was not the first company to produce a modem, or even a smart modem, the company has been richly rewarded for its relatively early entry of a quality communications product. It has continued its dominance through a combination of excellent support, product reliability, and highly visible marketing.

Are there any alternatives to Hayes? Yes and no. Several companies turn out less expensive Hayes act-alike products. These products come close to Hayes operationally, but often fall short on appearance, durability, and overall performance. Other products are available that could only be called one-up-on-Hayes modems. They add one or two outstanding—or at least improved—features to a generally Hayes-compatible performance, and a few dollars are added to the bottom line. Unfortunately, most communications software does not take advantage of those improvements.

For the present, the de facto Hayes standard has to be accepted. What does that mean for makers of board-level, medium-speed (600-9600 bps) modems? Compatibility with Hayes comes down to two factors: registers and commands.

REGISTERS

The Hayes Smartmodem registers are those that are designed by Hayes to control and report various communications parameters. All but a few bit-mapped registers can be read and modified by Smartmodem commands. The bit-mapped registers may be read, but they may not be written. Their values are bit-significant and report the status of the modem. Several of them are used almost exclusively for factory testing and are not intended for other uses.

The Hayes registers are named S0 through S17 and comprise the control interface between user programs and the UART (universal asynchronous receiver/transmitter), the real workhorse of the communications channel.

S0 **Answer on ring number.** This sets the answer off (S0 = 0) or to a specified number of rings (S0 = n). May be overridden by the A command. (Range 0 . . . 255, default = 0).

S1 **Count rings.** This is used when S0 > 0; it counts the number of ringing signals received and resets if no new rings occur within eight seconds of the last ring. (Range 0 . . . 255, default = 0).

S2 **Escape code character.** This stores the ASCII value of the command "escape" code. (Range 0 . . . 127, default = 43 ('+')).

S3 **Carriage return character.** This register stores the ASCII value of the key that is used to terminate command lines and result codes. (Range 0 . . . 127, default = 13 (CR)).

S4 **Line-feed character.** This stores the ASCII value of the line-feed character, which is output after the carriage return in *word* result codes. (Range 0 . . . 127, default = 10 (LF)).

S5 **Backspace character.** This register stores the ASCII value of the destructive backspace key (it cannot be a printable character). (Range 0 . . . 32 and 127, default = 8 (BS)).

S6 **Wait for dial tone.** This sets the amount of time in seconds that the modem waits after receiving the carrier before beginning the dialing sequence (two seconds minimum per FCC regulations). (Range 2 . . . 255, default = 2).

S7 **Wait for carrier.** This sets the amount of time the modem will wait to receive a carrier signal from the distant modem. (Range 1 . . . 255, default = 30).

S8 **Pause time for comma.** This sets the length in seconds of the pause produced by a comma command. (Range 0 . . . 255, default = 2).

S9 **Carrier detect time.** This sets the time in tenths of a second that a carrier signal must be present to be recognized as something other than data channel noise. (Range 1 . . . 255, default = 6).

S10 **Carrier loss to hang-up.** This sets the maximum time period of a temporary carrier loss that can be tolerated by the modem before the connection is dropped. (Range 1 . . . 255, default 7).

S11 **Touch-Tone timing.** This sets the duration of tones and the spacing between them in milliseconds for the Touch-Tone dialer. (Range 50 . . . 255, default = 70).

S12 **Escape code guard time.** This sets the time in increments of one-fiftieth of a second that must elapse before and after the escape command for it to be recognized as a legitimate command. (Range 20 . . . 255, default = 50).

S13 **Bit-mapped register (read-only).** The result code is set basic/extended, parity enabled/disabled, parity odd/even, data bits 7/8, buffer overflow, 8th bit stuck mark/space.

S14 **Bit-mapped register (read-only).** Local echo is disabled/enabled, result codes enabled/disabled, result code format digits/words, dial method pulse/tone, speaker ON until carrier detect, speaker always ON.

S15 **Bit-mapped register (read-only).** Originate/answer, duplex half/full, transmission rate, carrier ON/OFF.

S16 **Self-Test.** This controls the modem self-test feature (analog loopback test). (Range 0 . . . 2, default = 0: 0 = inactive; 1 = run self-test feature; 2 = tone amplitude test).

S17 **Bit-mapped register (read-only).** Modem ON/OFF hook, auxiliary relay contacts open/closed, autoanswer ON/OFF, speaker ON/OFF, speed high/low, ring indicator TRUE/FALSE, carrier present/not present, transmitter filter frequency (2400/1200 Hz).

COMMANDS

The commands, which are effective during local command state, may be used directly to control the modem. A special command called the *escape code* returns the modem to the command state when it is in the online state. By default this command is the character sequence +++ preceded and followed by one-second *guard time* in order to prevent normal transmitted data from causing an unwanted mode change.

The AT (attention) command tells the modem to execute what follows it as a command string. AT must be in uppercase letters. All other command words may be input in lowercase or uppercase letters. Each command sequence, up to 40 characters at a time, must have an AT prefix and is executed when a trailing carriage return is typed or issued by software. The only exception is the A/ command, which operates independently to repeat the last command that was executed.

Smartmodem commands that take an optional numeric value, *n*, default to 0 if no value is specified.

A **Answer.** Answers immediately. Tells the modem to answer regardless of the setting of register S0 and ring count in S1. Used to convert voice to data connections.

A/ **Repeat/Redial.** Repeats the previous command. (This requires no preceding AT or trailing carriage return.)

Cn **Transmitter carrier.** Controls ON/OFF state of the transmitter signal. (0 = OFF, 1 = ON).

, **Pause.** Pauses for a period specified in register S8 (two seconds is the default value).

D **Dial.** Originates a telephone call. No telephone or automatic calling unit is required. P or T is used to set either pulse or tone dialing method.

En **Echo.** Tells the modem to echo or not echo commands to the terminal screen. (0 = no echo; 1 = echo on.)

Fn **Duplex.** Selects half- or full-duplex operation. (0 = half duplex; 1 = full duplex).

Hn **ON/OFF hook.** Controls the simulated switch-hook status of the telephone connection. *On-hook* means the phone is inactive (effectively hung up) and *off-hook* means it is active. (0 = on-hook; 1 = off-hook; 2 = operate only line relay).

In **Identification.** Interrogates the modem for an identification (type) code. (Also retrieves a ROM checksum value from 1200-bps modems during factory testing.) (0 = interrogate modem for ID code; 1 = get high-speed modem ROM check sum (factory test only)).

Mn **Speaker (monitor) ON/OFF.** Several options allow the user to control the speaker in order to monitor the progress of a call. (0 = monitor OFF; 1 = speaker ON until carrier acknowledged; 2 = speaker stays ON).

O **Online.** Switches the modem from the local command state to the online state.

P **Pulse.** Enables pulse dialing (the power-on default). P and T may be freely mixed in a single dialing command.

Qn **Result codes sent/not sent.** Together with the V and X commands, this command controls whether or not result codes will be sent to the modem. (0 = result codes sent; 1 = result codes not sent).

R **Reverse.** Appended to a dialing command, allows a remote connection to an originate-only modem (reverses the local transmit and receive frequencies by putting the modem in the answer mode).

Sr? **Read register.** Reads the contents of register Sr and return the value (0–255) to a program or to the terminal screen.

Sr=n **Set Register.** Assigns the value *n* to register Sr.

; **Command state.** Returns the modem to local command state, following a dialing command, for example.

T **Touch-Tone Dialing.** Uses DTMF (dual tone, multifrequency) dialing method for subsequent dialing requests. (May be mixed with P requests).

Vn **Result codes as words or digits.** Selects the presentation of result codes (also see Q and X). (0 = result codes as digits; 1 = result codes as text strings).

Xn **Basic/extended result codes.** Extends the result code list to include 300/1200 bps indication. (0 = basic codes (0–4 only); 1 = extended codes (0–5)).

Z **Reset.** Resets the modem to default configuration. Clears anything following it in the command buffer.

When commands are sent to the Smartmodem, it attempts to perform the command. Assuming the registers have been set to have result codes returned, the modem returns either *digit codes* or *word codes*. The result codes indicate the success or failure of commands and the status of some operational parameters. In the following list all result codes are explained; the word result code is shown in bold.

0 **OK.** Command line was executed without error.

1 **CONNECT.** Basic set: carrier detected at 0-300 or 1200 bps. Extended set: carrier detected at 0-300 bps.

2 **RING.** Ringing signal detected.

3 **NO CARRIER.** Carrier lost (or never found).

4 **ERROR.** Error in command line—invalid command, command >40 characters, or invalid character format at 1200 bps.

5 **CONNECT 1200** (extended set). Carrier detected at 1200 bps.

—AH

Now you can satisfy all of your storage needs with one trip to Mountain. Upgrade your IBM,® AT&T,™ COMPAQ® and compatibles with Mountain's field-proven FileSafe™ disk, tape, and combo systems. All provide 100% compatibility with DOS 2.0, 2.1, 3.0, PC IX, UNIX™ and XENIX.™ And all come network-ready to support the 3COM Ether Series, Novell, Orchid PCnet,® NESTAR and STARLAN.™

THE HIGHER ROAD TO HARD DISK STORAGE.

For additional hard disk storage, choose from Mountain's high-performance internal and external FileSafe disks, featuring capacities of 10, 20, 30, 35, 68 and 120 megabytes.*

CLIMB-UP TO WORRY-FREE TAPE BACKUP.

If super-reliable tape backup is what you're looking for, our 27 and 60 megabyte internal, external or Mini Mountain FileSafe tapes set the industry-standard for fast, flexible backup of either IBM, AT&T, COMPAQ or Mountain hard disks.** Each provides full error correction (CRC) logic and a wide range of simple, menu-driven backup options.

*Every capacity not available for some computer models. Contact your dealer or call Mountain for details.
**QIC 24 interface; 450,000 installed in over 90% of all systems worldwide.

INDUSTRY STANDARD QIC-24 DATA FORMAT

® registered trademarks and ™ trademarks of Mountain Computer, Inc., American Telephone and Telegraph Co., International Business Machines Corp., COMPAQ Computer Corp., Microsoft Corp., Orchid Technology, and AT&T Bell Labs.

DISK AND TAPE COMBINED? WE LEAVE NO STONE UNTURNED.

If you need *both* high-performance disk and tape, our FileSafe combo systems combine all the features of FileSafe hard disk and tape in one low-cost, compact unit.

A ROCK-SOLID REPUTATION.

Mountain's hard-earned reputation among business users over the last eight years for responsiveness, innovation and reliability means we'll be there when you need us.

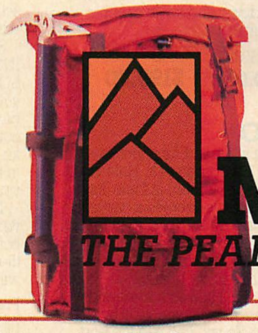
YOU'LL LIKE THE VIEW.

Whether it's disk, tape, or a combination of the two, look to Mountain for the industry's widest range of proven, high-performance storage upgrades.

For more information and the name of your nearest Mountain dealer, call:

1-800-458-0300
(in Calif., 1-800-821-6066)

CIRCLE NO. 154 ON READER SERVICE CARD



Mountain®
THE PEAK OF PERFORMANCE

Mountain Computer, Inc., 360 El Pueblo Road, Scotts Valley, CA 95066, (408) 438-6650, TWX 910-598-4504.
Mountain East Coast Office: (404) 441-7661. Australia: Data Peripherals 02-888-5733.
Canada: Parity Plus, Inc. (416) 673-3321. France: Compusol 01-530 07 37. Spain: Chip Electronica 03-201-2266.
Sweden: Nordic Data 08-730 50 50. West Germany: Distec Datensysteme 6172-23-081.

For a 28" x 36" poster (suitable for framing) of "Sunrise at the Grand Tetons in Wyoming," write Mountain Computer, Inc.

PC BRAND™: PROGRAMMING TOOLS FOR 'C' PROFESSIONALS

LIBRARIES: *Hundreds of Functions for Fast Programming*

C is a bare-bones language. It does not have built-in (machine dependent) functions for printing or floating point math or string manipulation. That's why it's so portable. That's also why libraries of pre-written functions are such big sellers: bullet-proof routines your application can effortlessly call to get the job done.

GREENLEAF FUNCTIONS

More than 200 functions in C and/or assembler source, and a 200 page manual with examples and demos.

32 DOS extensions: file and directory manipulation for DOS 1.1 and 2.0.

23 Screen Functions: Select mode, page, monochrome or color, palette; cursor shape, positioning; clearing and scrolling; character reading and writing with attribute and color control; pixel get and put; read light pen.

60 String Functions: Manipulation of strings including center and justify; efficient list operations which add, delete, and sort pointers to strings for top speed.

50 Graphics Functions: Primitives to access all graphics; typeface, formatting, and forms control for IBM Graphics Printer.

Plus keyboard status and function key assignment, time and date algorithms . . .we could go on!

Product Code: S0770 # Our Price:
List Price: **$185.00** **$140.00**

GREENLEAF COMMUNICATIONS

Want your application to communicate with other users or remote date bases? Now you can build asynchronous communications right into your C programs!

Over 60 functions set up an interrupt driven scheme with separate transmit and receive ring buffers (characters are simultaneously loaded at one end and transmitted from the other, or vice versa) for an arbitrary number of ports. Interrupt control means you can download a record, then halt the incoming stream to file it, display it, let the user tamper with it, send it back up line. It all happens within the context of *your* program, using your user commands, and does away with the sidestep to the separate communications software (and its semantics) you had to use until now.

The Greenleaf Comm Library supports ASCII or binary, any parity, any word length, 8250 UARTs, all four Lattice C memory models, Hayes 300, 1200, 1200B and other modems.

Its 80-page manual has examples of each function, discusses asynchronous communications, and both the library and demo programs come with source (a mix of C and assembler)!

Product Code: S0750 # Our Price:
List Price: **$185.00** **$140.00**

LATTICE C-FOOD SMORGASBORD

Decimal Arithmetic: Trigonometric, logarithmic functions, powers, conversion to strings, BCD operations for numbers up to 16 significant digits.

Level 0 I/O Functions: Direct operations for screen, keyboard, printer, and asynch port with no dependence on higher level I/O will minimize memory usage and maximize speed.

IBM PC BIOS Interface Access: Gets basic I/O services in ROM BIOS not available from the operating system to get and set video mode, cursor position, color and screen attributes, keyboard shift, scrolling, printer channel, port status.

Terminal Independence Package lets you move programs to computers with other types of terminals.

Product Code: S0200 # Our Price:
List Price: **$150.00** **$115.00**

Quite a treasure. So cost effective that after using only a couple of functions you'll be saving a bundle compared to writing your own. (It's true; how much do you cost per day?). So why not go for the whole bundle!

LATTICE WINDOW
All Your Applications Could Have Windows!

Windows are no gimmick. Ask any user of a windowed product.

But how do you add this technique to your applications without disappearing for several months of R&D?

With Lattice Window.™ A rare programming tool which will change the way you think of program design. Lattice Window takes over all screen management. It is an extensive set of object code functions which you merely call from your C program.

Think of a window as a screen of flexible size. You can tell Lattice Window to open and close up to 255 virtual screens from 1x1 to 255x255 bytes. Then tell "Window" to display any portion of these virtual screens on the physical screen — as many as fit — wherever you want, overlapping and overlaying at will.

Windows no longer needed may be closed. Any background area overlaid earlier (which could be corners of several windows) will pop back to the screen. Think for a moment what a programming job that technique represents.

To your program, the entered window is the *entire* screen — all row and column references are relative to the window no matter its absolute position in the screen. All scrolling occurs only within the active window. The cursor will not leave its boundaries until your program says to.

- You can move a window on the screen.
- You can grow it to display more of the underlying virtual screen. Or shrink it.
- You can control screen attributes (color, blinking, inverse video, etc.).
- You can direct output to either the monochrome or color board.

A formidable product to add incomparable flair to your next application!

Product Code: S0800 # Our Price:
List Price: **$295.00** **$235.00**

MULTI-HALO
MORE BOARDS!
Premier Graphics Library Now Supports 21 Boards

Halo has become so dominant a graphics standard that the newest incarnation, Multi-Halo, supports 21 graphics cards, and comes with a multitude of mouse and printer drivers. All with one purchase, replacing the previous one board per copy. The latest version adds support for the AT&T 6300 and the IBM enhanced graphics card.

Within your Lattice program a function call can reset drivers. You can arrange that your application program doesn't care what board the customer has; you can even switch boards within a program, as when two different monitors are used.

Halo is a complete library of graphic functions which can be linked with your Lattice programs to create full-color charts, graphs, simulations, animation.

Over 100 commands are at your disposal, including plot, line, arc, box, circle, plus single commands to produce bar graphs and pie charts. Pattern-fill and dithering commands give your graphics impressive texture and color mixes, rubber-banding draws shapes for interactive users, area moves produce animation, fill and flood commands paint areas. Multi-Halo allows you to define your own world coordinate scheme, divide the screen into windows, and scale graphic figures automatically.

Product Code: S0315 Our Price:
List Price: **$250.00** **$199.00**

PMATE
The Programmer's Word Processor

Pmate™ was designed for programmers. We'll wager that you cannot find a programmer who has discovered Pmate and moved on to something else.

Pmate is a full screen editor with ten auxiliary buffers for squirreling away pieces of text until needed. It uses single key commands to move the cursor, or text, or insert or delete, or rescue several thousand characters of deleted text.

It has a format mode for tab setting or wraparound and shaping when it's time to write documentation. Pmate lets you assign chains of commands or strings of text to single keys: a keystroke could set up the entire shell of a new C function, for example.

Pmate has variables, if-then statements, loops. It calculates, and converts decimal to hex to binary and back. You can write compact programs (called "macros") to delete comments, for example, or check syntax, or process long sequences of commands. Macros can alphabetize lists, do row and column math, perform a series of operations on multiple files, even summon other macros.

Put another way, Pmate is a text editor with its own built-in interpretive language. A language you can use to completely customize this text editor to your fancy. Possibly the most artful, ingenious program you have ever seen.

Product Code: S0600 # Our Price:
List Price: **$225.00** **$175.00**

Dr. HALO

Multi-Halo is a library of functions to utilize from your C program. Dr. Halo is a standalone program, written in Halo. Its dazzling. Your screen becomes a canvas, and a mouse (preferable) your paintbrush to take you wherever your imagination leads. Save the images to disk, run them by with Dr. Halo's slide show feature. Suggestion? Use Dr. Halo as a freehand sketchpad to design what you want your application to do. So you can see the end result you're after when you start to code.

Product Code: S0320 Our Price:
List Price: **$99.00** **$75.00**

BASIC_C
Use Your Knowledge of BASIC to Learn C

If you're getting the message that switching from BASIC to C would be prudent, you're about to discover that it's back to basics of a different sort. BASIC is fat with hidden functions that stripped down C just doesn't have.

Gone are all those handy string manipulators like LEFT$, MID$, STRING$, etc. In C, when you reach for even simple invocations like INPUT or PRINT A$,X% — well, underlying such expressions in BASIC are bulging macros which C cannot have if it is to keep its slim profile.

But now comes BASIC_C and all your old favorites are back. Someone has written the full set of C functions to mimic BASIC's vocabulary, from ABS to WRITE. Over 80 routines to open and close files, "field" file buffers, convert their contents from and to strings (the CV? and MK? series), peek and poke, print using, clear screen, "instr", on error goto. . .they're all there. Some have reworked names and syntax to suit C, but all are written as one-to-one functional equivalents to the familiar features of BASIC. And they are documented one to a page in alphabetical sequence like the Microsoft manual for added familiarity.

So with BASIC_C, when you're thinking INPUT, go ahead. Use it. Or LPRINT or LOCATE or INKEY. But without BASIC_C, you will find that every line of code plunges you back in the C texts to figure out how to write it. Someday you'll want to, but for now, BASIC_C will start you programming quickly at the statement level so that you can concentrate on C's larger concepts.

There's a bonus: an unusually well-written manual with a first rate chapter comparing how BASIC and C go about their tasks. Without question, BASIC_C will ease your transition to C.

Product Code: S0350 # Our Price:
List Price: **$175.00** **$145.00**

BASTOC
It Translates Your BASIC Programs Into C

You'd like to upgrade C, but what to do with that warehouse of BASIC routines you've carefully polished over the years? And how to face converting your famous Leviathan accounting system from BASIC, much less your Labyrinth file management package?

With BASTOC,™ a translator which takes in BASIC source code and emits pure Kernighan & Ritchie C. Our BASTOC understands Microsoft Extended BASIC (except for dynamic dimensioning and common). It will optionally convert your program into a single monolithic C function or will decompose it into separate functions, one for each GOSUB label.

Strings are dynamically allocated in the target program, ridding your application of BASIC's catatonic halts for garbage collection. BASTOC will try to create structure of even the most convoluted BASIC code, and writes any indigestible statement into the C output as a comment plus explanation. Also, you optionally can tell BASTOC to insert BASIC source lines into the C target as comments.

Source code of several modules of the translator and library are provided so you can tailor BASTOC to your environment if needed. And source of the entire product is for sale.

Product Code: S0370# Our Price:
List Price: **$350.00** **$280.00**

HIGH SPEED SOLUTIONS FOR APPLICATIONS DEVELOPMENT

BTRIEVE

Arguably the Best B-Tree File Handler Available

If your programming still gets anywhere near the files, here's our guarantee: Nothing will increase your productivity faster than handing the filing job to a sophisticated file manager.

Btrieve™ takes complete charge of all file indexing, reading, writing, insertion and deletion. Its 22 commands are library function calls which will accomplish everything you need to manipulate data files: create, open, and close files; delete and insert records, recapturing vacated space; find records which exactly or most nearly match keys; walk files by ascending or descending key.

Btrieve's foundation is a balanced-tree indexing scheme, conceded to be the fastest search technique devised (it will find any key in a million-plus item index in four or less accesses).

Btrieve comes with interfaces to C, and IBM Pascal, BASIC, and COBOL, and the manual gives you working sample programs which demonstrate every command in all four languages. The kind of presentation which led *PC World* to exclaim "for those of us who have endured poorly written. . . manuals, this one is a pleasure."

Btrieve has mainframe specifications! A single file may have up to 24 indexes. Segments of keys may be indexed. Each index can independently accept or block duplicate keys. A record length can be up to 4090 characters; an index length 255 characters. A file may be 4 billion bytes. It can even extend a file across two drives — even two hard disks!

With Btrieve you are freed to think logically; the physical file is no longer of concern. Gone for good is all that time-wasting dickering with intricate file referencing schemes and sorting algorithms.

Product Code: S0650 — Our Price:
List Price: **$245.00** — **$199.00**

PANEL — REVISED!

Version 6 Adds the 3rd Dimension

The newest version of this premier programming tool lets you layer your screen designs with up to ten overlapping images, making it easy to background a screen with pop-up lists, help boxes, and alternate sets of input fields.

Writing your own screenware is a good way to blow completion dates and profits. Panel™ works with you interactively to set up foolproof screen displays and data entry forms rapidly. It tests your form to prove that it (and test data) behave correctly, then converts the finished work into C source code for incorporation into your application, and compilation with Lattice.

Wonderfully diverse attributes may be selected for any field — size, data type, color, of course, but also: conversion of input to upper case; clearance of existing data when new entry is started; masks for standard formats (e.g., dates, phone numbers); a choice of styles for numeric fields; phrases which fill in when their first letter is typed; multiple-choice lists from which to choose a field fill-in by cursoring a highlighted bar. Fields may be multi-lined (e.g., name and address as one field) and scrolled if larger than the screen space allotted them.

Panel builds in a user interface for keystroke movement within and between fields, and supplies extensive validation routines for checking user field entries — in source code, so you can tack on your own unique variants.

Panel's screen layouts can be assigned to individual windows under Lattice Window control. Screen designs may be dynamically loaded from file, or compiled into a program, and version 6 has optimized code to quicken display speed.

The whole package is wrapped in a monitor and keyboard customization package to tailor* your application for other equipment.

Panel. A superior productivity tool now bigger than ever.

Product Code: S0400 #* — Our Price:
List Price: **$295.00** — **$235.00**

*"Tailor" needs separate license to distribute with your programs

PLINK86

Size Unlimited with This Overlay Linker

Software is becoming ever more sophisticated, which means more complex programs needing large chunks of memory. But if you use extra memory, you are counting on users to have expanded RAM, and will forego sales to those who do not.

Plink86™ is the answer. It shoe-horns large programs into small memory. First, Plink86 acts as an alternative to DOS' Link. C encourages design of separately compiled object modules in the Microsoft relocatable format. Plink86 pulls modules together into single compiled programs. But Plink86's overlay power is what has gained it a reputation as a miracle worker. It binds into the compiled program its overlay manager which knows how to swap modules of your large linked program between disk and memory, so that each can temporarily occupy the same memory space.

Unlike other linkers, the overlay manager acts on its own, needing no calls from the source program. Instead, Plink86's straightforward overlay description language allows you to describe your overlay structure in one place in your program — a structure permitting up to 4,095 overlays stacked 32 deep. No re-compiling to re-arrange the structure!

Plink86 can even sub-divide its linked output into multiple files for programs which must span more than one disk.

PC Tech Journal calls it "the premier linker and overlay loader that's taking over at many of the programming shops around the country."

Product Code: S0500 # — Our Price:
List Price: **$395.00** — **$315.00**

LATTICE C COMPILER

Over 25,000 Users Make it the Best Seller by Far

Lattice C has been the standard-bearer compiler for years. Peter Norton in *PC Magazine* (1/8/85) said, "best for systems programming. . . tight code. . . total control of what's going on. . . noticeably better than any of its competitors". After reviewing nine compilers for the PC, the *PC Tech Journal* unequivocally declared Lattice C "best for software development. . . it compiles fast and produces fast programs."

When Lattice C appeared, there was nothing comparable. Ever since, software developers have created exceptional tools and function libraries to marry into Lattice C. Programmers now have an enormous resource of libraries and utilities intended for Lattice C to speed their work. That's a vital consideration in choosing a compiler. Better check whether the latest geewhiz may have only gewgaws to go with it.

Lattice C runs on virtually any computer using an 8088 or 8086 chip, and generates code which optimizes use of the 8087 chip, if present. Create your source files with any word processor or text editor and Lattice C will compile them into Intel 8086 object module format for linking to other modules by DOS's Link or our Plink86.

Lattice C offers a choice of four memory modules between 64K and RAM capacity for program and data to allow you to choose the right combination of small memory efficiency and large memory addressability. It is a full implementation of Kernighan and Ritchie (K&R), with 39 character variable names, nested comments, transcendental and Unix math functions K&R didn't think to mention, and some of Unix's most useful options such as "fork", to pull a second program into memory alongside the first, branch to it, and return.

Product Code: S0100 # — Our Price:
List Price: **$500.00** — **$255.00**

CDEBUGGER NEW!

MSD's Source Level Pesticide

We've offered "symbolic" debuggers for some time. They can refer to program symbols — the names you give to variables, functions, etc. in your source programs — so you don't have to debug C programs by way of hex memory addresses. Now along comes a true source level debugger for programs compiled with Lattice — and at a breakthrough price.

Switch on the Cdebugger™ source display, from the beginning or at any breakpoint, and enjoy the uncanny experience of watching your source code step by, one line at a time. Or skip ahead n lines and display from there. Or browse through source from the surrounding program area — even from another file.

Multiple breakpoints can be assigned to individual lines in your source files. Areas of memory can be set read-only to let a form of write protection help find the most elusive bugs, like pointer encroachment. Trace options allow display of function names and line numbers as executed. Backtrace writes a history of all functions called, a dandy roadmap of how you got lost.

Other commands display and alter memory and registers, show and replace expression values. Simple variables may be referenced directly by name. (Not local names which the compiler doesn't map.)

No list of source files need be provided to Cdebugger. It simply targets all .c files in the current directory. Cdebugger supports and automatically recognizes all four memory models. Needs 256K.

Product Code: S1100 # — Our Price:
List Price: **$165.00** — **$135.00**

RUN/C

A C Interpreter at Last — And it Works like BASIC!

C is contemplative, elegant. But even elitists yearn (silently) for the simpler days of throwing together code with the instant interaction of an interpreter.

The good old days are here again. We have not just a C interpreter, but one that acts like BASIC! Use RUN/C's in-line editor to create a program. RUN it, just like BASIC. If it stumbles, just LIST it, EDIT it, add lines, delete lines, RUN it again, fix it again. The simple luxury of break key interrupt is restored, and RUN/C™ even has TRON, TRACE, and a profiler which reports the number of times each program line is executed.

RUN/C's "editor" command will save a program and automatically hand it to your favorite editor for further work. Or you can use its "shell" command to summon your compiler from *within* RUN/C.

For the professional, RUN/C is ideal for rapid program development. Put up code at high speed, tinker and re-arrange, throw in test printouts, bang away at tight little sections until they're bullet proof — *then* hand them to the compiler.

RUN/C has a treasury of functions built into the interpreter — over 100 paralleling the most used functions found in standard compiler libraries. Only programs calling other functions or other libraries for which you do not have source need special measures (e.g., comment-out, dummy functions). Well worth it to win the productivity gains of an interpreter.

Your disk supplies a separate source code file demonstrating each of RUN/C's library functions, with a manual to match. In fact, if you are new to C, this product doubles as a superb tutorial! (Requires 256K, DOS 2.x).

Product Code: S0900 # — Our Price:
List Price: **$150.00** — **$109.00**

RUN/C, BASIC__C, and Bastoc **$479.00**

TERMS AND CONDITIONS OF SALE

Licenses: Each price is for a license to use a product on a single computer and does not constitute product ownership. Products coded "#" may be used to create programs for distribution without royalty payments or additional licenses, provided said programs do not substantially replicate the products themselves.

Compatibility: PC BRAND's standard products are designed to operate with the IBM PC, XT or AT under PC-DOS and require no more than 128K of RAM unless indicated.

Returns: Products returned will be accepted for refund or credit only by prior authorization. Software diskettes are delivered in sealed envelopes and are unreturnable if seal is broken. Defective products will be replaced in whole or part.

Payment: We honor American Express, MasterCard & Visa (no surcharge), checks in advance, COD for cash or certified check, and funds wired to Chemical Bank, 126 East 86 St., New York, Account: 034-016058. Purchase orders accepted from larger corporations and institutions at our discretion; 2% per month added to balances unpaid after 30 days.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3.00. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express: 1st product $18, each add'l $6. Non U.S.: $15 per order for export/import forms; variable shipping charge.

What're you waiting for

800-PC BRAND

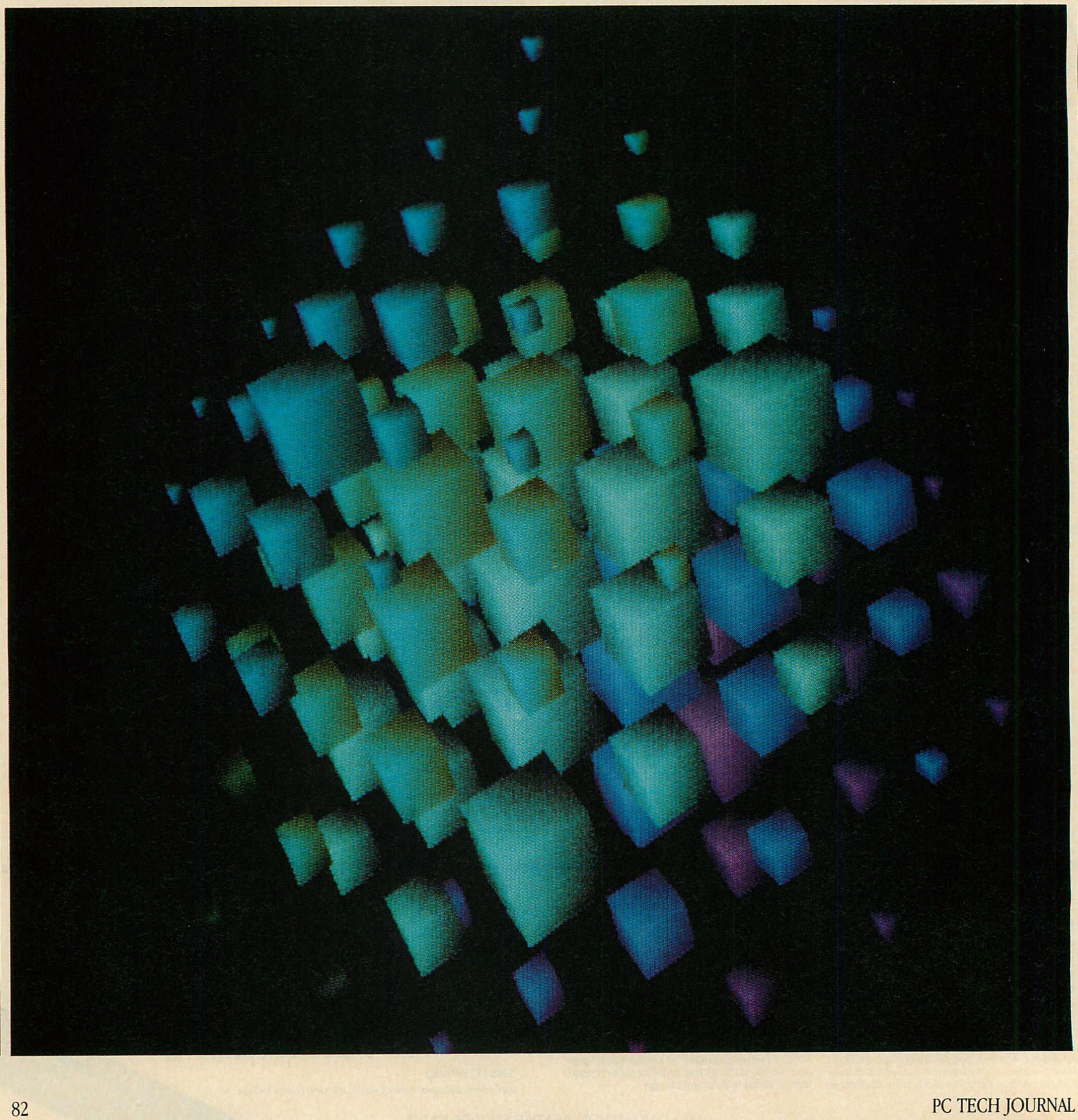
That's (800) 722-7263. In NY State call (212) 410-4001.
PC Brand, P.O. Box 474, New York, N.Y. 10028
Telex: 667962 (SOFT COMM NYK)

Prices, terms, and specifications subject to change without notice.

SAVE $5.00. We'll take $5 off your order if you tell us this code: T9

DATA MANAGERS AS DEVELOPMENT TOOLS

A Data Manager with Kernel Code Generation



COMPUTER GRAPHIC • DOV JACOBSON

Although Microrim's R:base 5000 does not do all the programming, it provides applications developers with a rough sketch.

STEVEN ARMBRUST and TED FORGERON

The intense competition in the development of personal computer software is nowhere more evident than in the field of database management products. Microrim is trying hard to unseat the dominant product, Ashton-Tate's dBASE series.

R:base 5000, the newest entry from Microrim, is the successor to R:base 4000 and the company's answer to dBASE III. Microrim claims that R:base 5000 is one of the most powerful, easiest to use, and fastest database products in the PC market. The accompanying sidebar from Data Decisions (a *PC Tech Journal* sister company) lists the vital statistics for R:base 5000.

This article is the second in a series on database managers. It will examine R:base 5000 by building a sample database devised by the editors of *PC Tech Journal*, the same one used in all of the articles in this series. (For a complete explanation of the sample application, see "Sample Application Specifications," August 1985, p. 48. The article also is available for downloading on PCTECHline, 301/576-PCTJ.) This permits a straightforward comparison between the features of and benchmark results for R:base 5000 and all the other products reviewed.

Microrim's claims for R:base 5000 are simply stated. The company calls it a relational database manager that is sophisticated enough to satisfy serious developers, easy enough to use so that even the nonprogrammer can create a powerful database, and fast enough for serious business applications—in short, a top-of-the-line product without the complicated interface that is often associated with such power.

To achieve these seemingly disparate goals, Microrim has produced R:base 5000 in several pieces: Application EXPRESS, a source code generator; the R:base interpreter program; FileGateway, a data transfer program; RBEDIT, a source code editor; and the RBCOMPILE compiler. All of these items are included in the standard $700 R:base 5000 package.

Application EXPRESS is a streamlined method for setting up the structure of the database (defining the fields and records), creating a hierarchy of menus, producing forms for data entry, and setting up simple reports that extract information from the database. Instead of requiring individual commands to perform these operations, Application EXPRESS is a series of menus that guides the user through the creation process. This helps nonprogrammers set up simple database systems on their own.

Any task performed by Application EXPRESS also can be done directly with R:base 5000 commands (in fact, Application EXPRESS generates a file of R:base commands). The R:base commands are part of a complete procedural language that provides much more power than Application EXPRESS. Users can start with a system built completely with EXPRESS; then, as they become more familiar with the product and want more complex features added, they can create procedures using R:base commands and link those procedures to the original application or add code to the source created by Application EXPRESS.

R:base 5000 also contains special form-building and report-writing functions that allow users to build more complicated forms and reports than are possible with Application EXPRESS. As with procedures, these forms and reports can be linked into the application using Application EXPRESS.

FileGateway is R:base 5000's data-transfer program, formerly available only with CLOUT, Microrim's natural language query program. FileGateway allows users to read Lotus 1-2-3, pfs:file, dBASE II, Multiplan, DIF, or ASCII files, and transform the information into R:base 5000 format. As with Application EXPRESS, the user communicates with FileGateway by choosing from menus and answering questions.

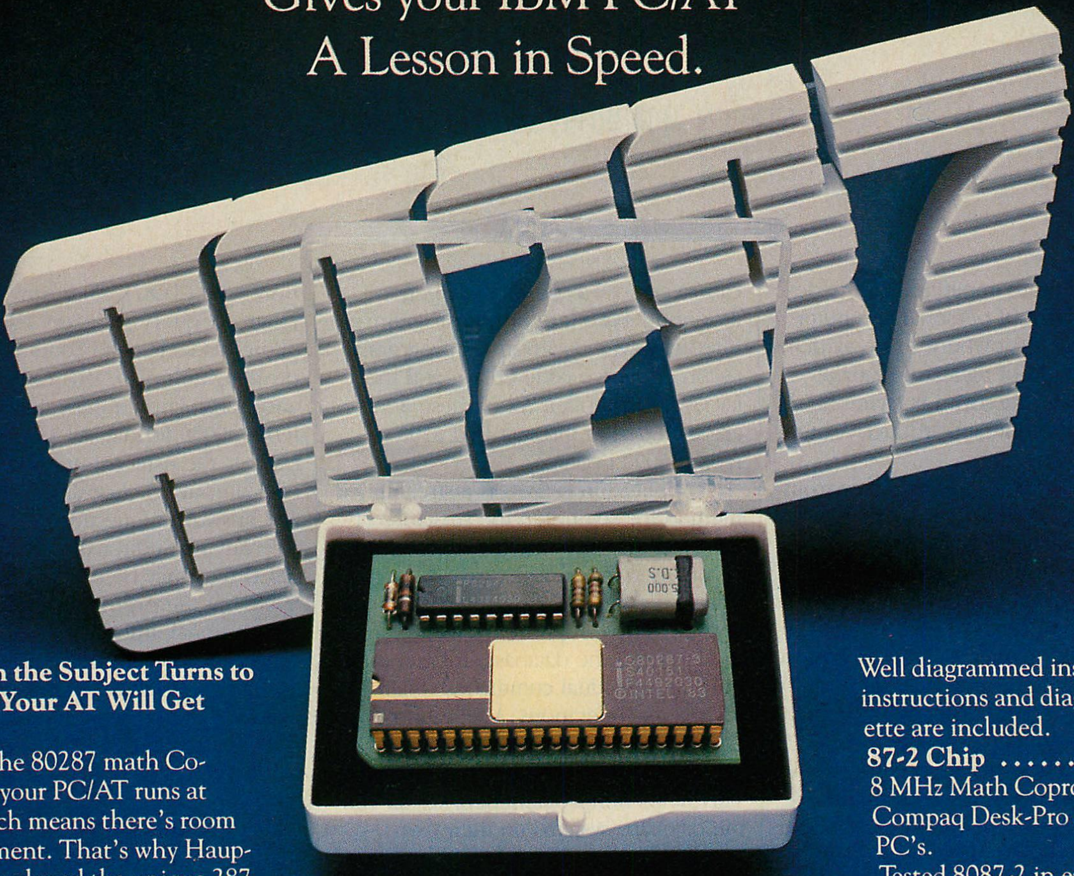
Any text editor that produces pure ASCII files may be used with R:base 5000, but the product comes with a conventional, full-screen text editor called RBEDIT. With this editor, users can build files of R:base commands to be executed at one time. If another editor is used, care must be taken not to include Tab characters, because R:base 5000 cannot process command files with Tabs. This can be an inconvenience for users of editors like IBM Personal Editor, which includes Tabs unless specifically directed not to.

The last item included in the R:base 5000 package is a compiler that transforms the ASCII program files into a format that can be interpreted by the R:base program. The advertised benefit of RBCOMPILE is security; with binary files, unauthorized users cannot tamper with the data or change the commands. Compiling might also improve performance. A compiled file is partially interpreted, and program blocks are combined into a single file, limiting the number of files that must be opened and closed. Microrim does not stress this point, nor does it guarantee better performance after compiling.

Optional items that are available for use with R:base 5000 include CLOUT, Program Interface (which allows Pascal and FORTRAN programs to access a database created by R:base), and Extended Report Writer (primarily for R:base 4000 users because R:base

HAUPPAUGE

Our 287 FAST/5
Gives your IBM PC/AT
A Lesson in Speed.



Now, When the Subject Turns to Fast Math, Your AT Will Get an A+.
Right now, the 80287 math Coprocessor in your PC/AT runs at 4MHz. Which means there's room for improvement. That's why Hauppauge has developed the unique 287 FAST/5—a module that gives your AT an unmatched 5MHz capability. Speeding up your math computations by 25%. Now that's an A+ performance!

• Easy Installation.
Inventors of the 87 chip-in-a-carrier, Hauppauge has created a carrier mount for their 287 FAST/5 too—insuring easy installation. Just plug the 287 FAST/5 into your PC/AT's 80287 Numeric Coprocessor socket. No expansion slot required.

• Low Price.
The 287 FAST/5 is comparable in price to the standard 80287 chip.

287 FAST/5 .... $295
Hauppauge's 80287 Math Coprocessor enhancement module—boosting operation speed in the PC/AT to 5MHz—includes well diagrammed installation instructions and a diagnostic diskette.

87 Chip $149
Our 8087 Math Coprocessor chip is carrier mounted for easy installation in the PC, PC/XT and PC compatibles. Well diagrammed installation instructions and diagnostic diskette are included.

87-2 Chip $249
8 MHz Math Coprocessor for Compaq Desk-Pro and Olivetti PC's.
Tested 8087-2 in easy-to-install chip carrier with installation instructions and diagnostic diskette.

287 Chip $295
80287 Math Coprocessor chip. A 5 MHz chip that runs at 4MHz in an IBM PC/AT. Carrier mounted for easy installation. Includes installation instructions and diagnostic diskette.

87 Software Pak $180
We offer math interface libraries for the IBM BASIC, PASCAL, and FORTRAN Compilers, and for the IBM MACRO Assembler. Plus, a set of high speed Matrix manipulation routines. All libraries include complete source code and a one year free software update service. (Special offer: the 87 Software Pak is only $120 when purchased with the 287 FAST/5 or any of our chips.)

Hauppauge Computer Works Inc.
358 Veterans Memorial Highway, Suite MSI,
Commack, New York, USA 11725 • 516-360-3827
Available from your local computer dealer

Advertising: Scali Communications, San Francisco

Hauppauge
(Pronounced: "Ha-pog")

CIRCLE NO. 164 ON READER SERVICE CARD

5000 already contains most its features). None of these options is included in the base price of R:base 5000.

R:base 5000 has no copy-protection, so installation is easily accomplished by making working copies of the R:base diskettes or copying the information to a hard disk. Although diskette-based systems are supported, R:base 5000 is a full-featured product that works best if a hard disk is present.

After installation on a hard disk, typing the command RB5000 brings up the main R:base menu, which includes the choices of FileGateway, Application EXPRESS, the R:base program, RBEDIT, and RBCOMPILE. (A diskette-based system requires each program to be invoked separately.) As with all R:base 5000 menus, pressing the Space bar moves the cursor from item to item, and pressing Enter selects the item.

If the user chooses the R:base program from the main menu, the normal R:base R> prompt appears. From here he can enter commands. Because the R:base 5000 language is interpreted, commands can be entered directly, or an entire command file can be run using the RUN command.

The R:base prompt changes to denote different operations within the program, such as database definition, help, and data loading.

DATABASE DESIGN

R:base 5000 uses the traditional relational database terminology: a file is called a *table* (tables are not maintained as separate files, but are all combined into a single file); a record is called a *row*; and fields are called *columns*.

Like most database managers that run on microcomputers, R:base 5000 is a relational database, meaning that no implicit relationship (such as parent/child) exists between the files in the database. Instead, the files are related by having a field in one file that is the same as a field in another file.

One of the advantages of R:base 5000 over other database management packages is the ease with which the file structure can be added to or changed. R:base 5000 offers commands to rename columns and tables, add or delete columns, or even (within reason) to change a column's values from one data type to another. Many of these operations also can be performed using the simple interface of Application EXPRESS, but this applies primarily to operations on empty databases.

Relational databases rely on identical columns to relate items in one table to those in another. In order to aid in maintaining database integrity, R:base 5000 has commands that allow users to program changes in corresponding items from different tables.

Many relational algebra commands are implemented in R:base 5000 so that existing tables can be combined in a variety of ways. PROJECT creates a new table that is a subset of an existing table, one that includes only specific columns in a specific order. APPEND takes rows from one table and adds them to the end of another table. UNION combines the rows and columns of two tables into a single table. It matches up the rows by comparing the values in columns with identical names. INTERSECT works just like UNION, except that it includes only those rows that have a common column with a matching value.

The R:base commands are part of a complete procedural language that provides much more power than does Application EXPRESS.

SUBTRACT is the opposite of INTERSECT. It finds the rows in one table that do not match up with those in another table. JOIN is similar to UNION in that it forms a new table using the rows of two existing tables. However, it includes the rows in the new table based on a specified comparison of the values in two rows (even using rows of different names). For example, if one table contains the column LASTYEAR and another the column THISYEAR, JOIN could produce a table of combined rows in which each THISYEAR value was greater than the corresponding LASTYEAR value.

With these relational commands, data can be kept in small tables and combined later to produce reports or specific screen displays.

R:base 5000 supports six different data types: DATE, TIME, DOLLAR, INTEGER, REAL, and TEXT. No arrays or extended types are supported.

Each DATE can be displayed in several different formats, depending on the setting of the command SET, which determines certain global operating conditions. (Another operating characteristic can decide whether sorts are done strictly by ASCII code, or whether uppercase and lowercase characters are sorted together.) TIME is typically displayed as hh:mm:ss. R:base 5000 has functions available to retrieve the current date and time.

DOLLAR data types appear with a dollar sign and a decimal point whenever they are displayed. They can handle values in the range of ± $99,999,999,999,999.99, which is enough for most businesses.

The INTEGER data type supports integers in the range of ±999,999,999. REAL numbers have a range of ±9 * 10 ± 37 with six-digit accuracy. TEXT fields can consist of 1 to 1,500 characters. When defining a database, only TEXT columns have a character limit. All other data types are calculated to their fullest precision, limited only by the space on a specific form or report.

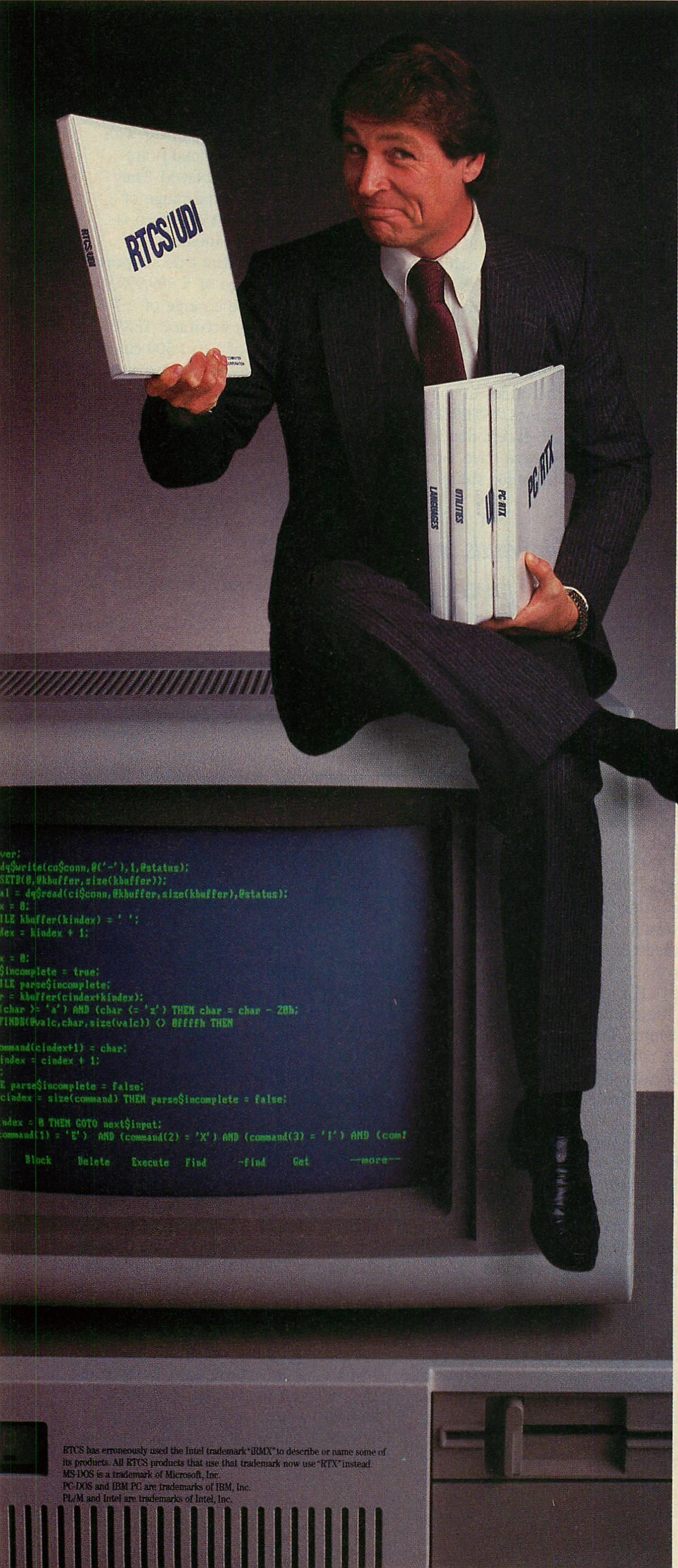
Internally, each DATE, TIME, REAL, and INTEGER value requires four bytes. A DOLLAR value takes eight bytes. A TEXT value requires the number of bytes equal to the column width, plus one extra if necessary to round up to an even number of bytes.

Each database defined with R:base 5000 can contain as many as 40 different tables. The size of the entire database is limited only to the maximum DOS file size. Within a table, 400 columns can be defined, with a maximum row size of 1,530 characters.

Once the data are entered, R:base 5000 maintains the entire database in just three files. The first is a directory that lets R:base access all the tables of information. This file is fixed at 9,600 bytes. The second file contains all the data in the database—as much as 40 tables worth of information. This file can easily exceed the size of a floppy disk, so a hard disk is necessary for most applications. The third file is the set of key indexes; they speed up access to a particular record in a file or to the entire file in key sequence.

Microrim promotes its Application EXPRESS as a productivity tool, especially for novices or nonprogrammers. Programmers, too, can benefit from EXPRESS, especially in setting up the database structure and creating menus.

Invoking EXPRESS, either by typing **express** at the DOS prompt or by selecting it from the main menu, displays the EXPRESS menu. The **Define a New Database** option displays a form similar to that shown in photo 1 (although the blanks in this form already are filled in). Setting up the structure of a table involves simply naming the table, naming each column in the table, and assigning each column a data type and, for TEXT types, a length. The information shown in photo 1 is the structure

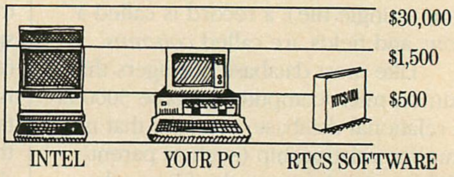


Extend the capabilities of your PC/MS-DOS™ based system and duplicate the Intel™ Series III, IV, and 86/310 with RTCS software development tools.

Don't buy expensive hardware to get the exact same professional development capabilities. For around $500, the RTCS/UDI Universal Development Interface software lets you use Intel Series III Microcomputer Development System software, available from RTCS, on your IBM PC™ or any other MS-DOS-based computer. And Intel software runs up to 20% faster on your PC using the RTCS/UDI.

DID YOU KNOW YOU'RE ALREADY SITTING ON A $30,000 INTEL SYSTEM?

Fortune 500 companies like IBM, General Electric, Honeywell, and over 1,500 others have found the RTCS/UDI easier to use with higher performance than most Intel systems. Besides the obvious expenditure comparison, they see Intel Systems as special purpose computers with limited software. But with RTCS, a PC/MS-DOS-based computer becomes a reliable performer with a wide repertoire of software possibilities.



Also available from RTCS is PC/RTX, Real-Time, Multi-user Multi-tasking Operating System for IBM PC and others; Intel languages such as Pascal, Fortran, PL/M,™ and C Compilers; Utilities such as high level software debuggers, as well as a wide range of software support supplied with the RTCS/UDI at no additional cost.

Before you go out and buy expensive hardware for capabilities you're already sitting on, call RTCS today. **(805) 987-9781**

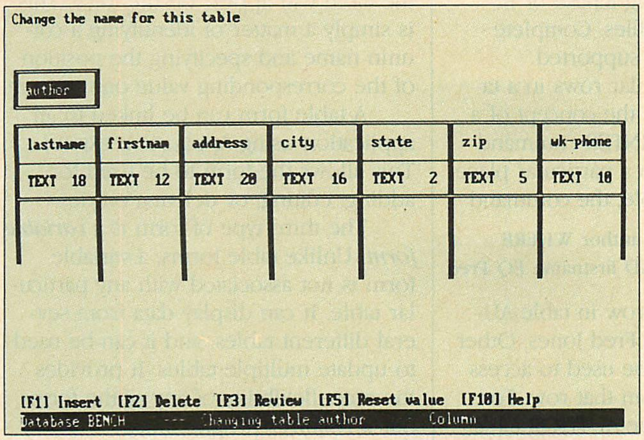
CIRCLE NO. 138 ON READER SERVICE CARD

RTCS

Getting the most from what you have.

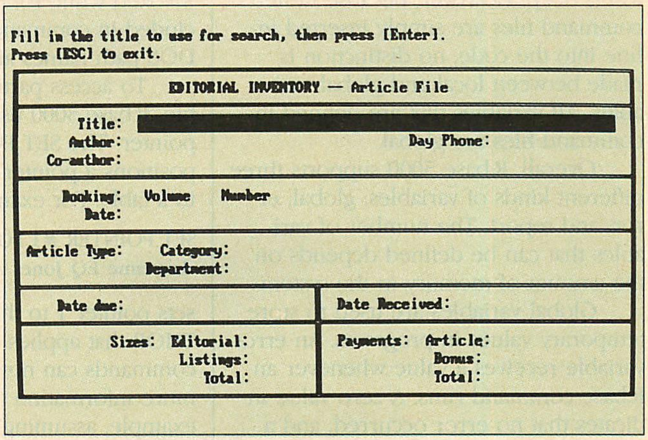
1390 Flynn Road, Camarillo, CA 93010
Telex: 467897 RTCS CI

PHOTO 1: *Table Structure*



With Application EXPRESS, setting up the structure of a database is as simple as filling in a form. The program prompts the user for all the information.

PHOTO 2: *Sample Data Entry Screen*



The FORMS command was used to design this screen, but custom programming, shown in listing 1, was required to make it accept data from multiple files.

of one of the tables used in the *PC Tech Journal* sample application.

Application EXPRESS simplifies the process of defining multiple tables with related elements, because whenever the same column name is re-entered in setting up a different table, EXPRESS remembers its data type and length and automatically fills in that information. A similar process changes the structure of a database. It works fine as long as the database contains no data. Once data are loaded in, Application EXPRESS has reached its limit. Only R:base commands can change the structure then.

The other particularly useful feature of Application EXPRESS is its ability to set up menus for a user application and to link those menus to user-defined code. Creating a menu involves following the instructions that appear on the screen. The only drawback is that the menus defined with EXPRESS support only two formats: horizontal (a line of options displayed across the top of the screen) or vertical (a complete screen with menu items laid out vertically). EXPRESS can set up menus in minutes.

Once a menu is defined, Application EXPRESS lets the user choose an action (including help messages) for each one of the items on the menu. As expected, two of the actions are to show another menu or to exit to a previous menu, features that allow users to build hierarchical menu structures.

EXPRESS also lets users create forms for loading data into a table, examining and editing table information, deleting rows, and building and printing reports. Further, it builds the code to perform the selected operations.

At first glance, this last capability may seem to be a dream come true. Application EXPRESS does some necessary programming. However, its ability to create complete applications is quite limited: forms built using EXPRESS are limited to a single format; data can be read from or written to just a single table; user input cannot be used as a criterion when searching for a row to examine; and the format and type of information in reports is limited.

These are not limitations of the entire product; rather, they affect only those functions that Application EXPRESS can do by itself. EXPRESS also has the ability to link menu choices to user-written code or to forms and reports created by the product's other form-building and report-building tools. It can link together up to 42 different *blocks*, where a block is defined to be a menu, any other screen display, or a command file (a piece of user-written code). This ability allows anyone to wrap all database routines into a single, easy-to-use menu structure, giving the applications a professional feel.

Application EXPRESS allows users to define help screens for each menu; a maximum of 24 lines of text are permitted. Later, when the application is running, the user can obtain help simply by pressing the F10 key.

When the user exits from Application EXPRESS, the program translates all of the user's specifications—menus, forms, and reports—into standard R:base commands that it places in a separate file. Even the user code that it links in is pulled into this file. To run the application, the end user simply invokes this command file.

The command file is written in ASCII and can be edited to add or change features. However, once this file is changed manually, the user cannot return to Application EXPRESS to make any more changes to the application.

A word of caution must be emphasized here. Although Application EXPRESS links menus to user-written code, it copies that user-written code into a separate file. Therefore, if any changes are made to the user routines, the routines must be relinked using Application EXPRESS, or the changes that were made will not take effect.

THE LANGUAGE

Despite Microrim's claims about the ease of using Application EXPRESS, developing all but the simplest applications will require some programming. The R:base 5000 programming language offers many of the features contained in high-level languages such as BASIC and Pascal, but it has some limitations too.

The control and branching capabilities of R:base 5000 are very good. It offers IF..THEN..ENDIF structures, WHILE..ENDWHILE structures, and a GOTO command. IF and WHILE structures can be nested up to ten levels deep. However, the command structures contain some seemingly arbitrary limitations. For example, a maximum of 4,000 characters are allowed between WHILE and ENDWHILE statements.

R:base 5000 is also deficient in its subroutine capabilities. Subroutines exist in name only; they are really separate command files. Any group of R:base commands can be stored in a command file and invoked with RUN. R:base considers subroutines to be command files that are called by other command files.

R:base has a provision for passing as many as nine parameters (using the

familiar %0 through %9 notation), and command files can be nested up to five levels deep. However, because nested command files are simply inserted in-line into the code, no distinction is made between local and global variables. All variables that are defined in command files are global.

Overall, R:base 5000 supports three different kinds of variables: global, error, and report. The number of variables that can be defined depends on the amount of memory in the system.

Global variables are used to store temporary values in programs. An error variable receives a value whenever an R:base command runs. A zero value indicates that no error occurred, and a nonzero value indicates an error. An appendix in the manual lists error conditions and their values; unfortunately, the list is ordered functionally instead of numerically or alphabetically. To perform error checking, the programmer can use control statements to test the error variables. A program can even take specific action based on the value of the error variable.

Report variables are used when defining reports. Like global variables, these are used to store intermediate values in calculations or final values to show on the report.

Although R:base 5000 provides the ability to perform mathematical calculations using standard arithmetic operators, it does not include a large group of ready-made mathematical functions. The COMPUTE command is available to compute averages, maximums, minimums, the number of rows in a table, the number of entries in a table for a particular item, and the sum of all numerical items in a table. However, R:base 5000 has no financial functions such as Lotus 1-2-3 provides.

String processing is achieved simply. TEXT variables can be combined using the + operator (with no spaces in between) or the & operator (with a single space between). Other text can be inserted by enclosing it in quotes. A MOVE command is provided to retrieve a portion of text from a single TEXT variable and insert it at a particular place in another text variable.

R:base 5000 does not have a direct interface to DOS commands or system calls, nor does it allow the user to suspend R:base, invoke another command, and resume. However, it does include several commands that are identical in name and function to the equivalent DOS commands: CHDIR, CHKDSK, COPY, DIR, ERASE, MKDIR, RENAME, RMDIR, and TYPE. Also provided is a CHDRV command to set or display the default drive. These commands can be invoked while running R:base or included in command files. Complete DOS path names are supported.

To access particular rows in a table, R:base 5000 uses the concept of a pointer. The SET POINTER command positions a pointer to a particular place in a table. For example, the command

```
SET POINTER #1 FOR author WHERE
 lastname EQ Jones AND firstname EQ Fred
```

sets pointer 1 to the row in table AUTHOR that applies to Fred Jones. Other commands can now be used to access more information from that row. For example, assuming the AUTHOR table also has a column called HM-PHONE to store the phone number, Fred Jones' phone number can be stored in a variable with the command

```
SET VARIABLE var1 TO hm-phone IN #1
```

Other commands are also available to change or delete information in the current row or to move the pointer to the next row. As many as three pointers can be used at the same time.

The number of variables that can be defined in R:base 5000 depends on the amount of memory that is available in the system.

OPERATING CHARACTERISTICS

R:base 5000 provides three different kinds of forms, each progressively more difficult to create, but each with progressively more power. The first form is simple to create using Application EXPRESS. Its layout is fixed at one field per line in the order that the fields appear in the table definition. No special graphics characters can be included, and it applies to one table only.

The second kind of form, called a *table form*, also applies to a single table. It is created using the FORMS command, which requires two phases. First, an editing pass occurs in which the user types in all the fixed text of the form. Graphics characters can be included using the combination of the Alt key and the numeric keys to specify the ASCII values. The text of the form can be placed anywhere.

After the form has been edited, the user can locate where the values from the table will appear on the form. This is simply a matter of identifying a column name and specifying the position of the corresponding value on the form.

A table form can be linked to an application using Application EXPRESS. This allows the form to be used for adding, editing, or deleting entries.

The third type of form is a *variable form*. Unlike table forms, a variable form is not associated with any particular table. It can display data from several different tables, and it can be used to update multiple tables. It provides the most flexibility of any of the forms, but it requires programming.

The FORMS command also is used to set up variable forms. Instead of assigning columns of a table to the form, variables are assigned to particular places on the form. A program must be written to assign values from tables to these variables. Special commands, such as DRAW, ENTER VAR, and EDIT VAR, can be used to display the form on the screen, enable the operator to change the fields, and update the fields with new information.

R:base 5000 has a number of operating characteristics that can be set at a global level using the SET command. For example, the program can be made to beep when an error occurs, foreground and background colors can be set, echoing can be turned on or off, error messages can be displayed, the number of lines of data displayed at one time can be set, and the width of pages sent to the printer can be set. The program can be set to write to disk after every update or keep the changes in memory until the user exits the program. The format of DATE fields can be changed (for example, from month-day-year to year-day-month). The user also can decide to distinguish between upper- and lowercase characters in sorts and other expressions. The Esc key can be activated or turned off as a method of aborting a command file.

Like many PC programs, R:base 5000 can take advantage of the extra power of the 8087 coprocessor. If the computer has an 8087, R:base will use it for real-number calculations. Otherwise, software is necessary.

R:base 5000 provides several tools to aid in debugging command files. First, error messages can be turned on while testing code. If errors occur, the text of the error message is displayed on the screen.

The command SET ECHO = ON causes the commands in a command

file to be displayed as they are run. Combining this command with PAUSE and SHOW VARIABLES allows the programmer to step through code and examine variables along the way.

No runtime package is yet available for R:base 5000, but Microrim plans to introduce one later this year. (A runtime package is a low-priced subset of a database manager that provides the facilities for running applications, but not all the tools for developing them. Typically, a programmer or database developer would use the full package to develop an application and supply that application along with the runtime package to end users.)

The R:base 5000 package itself does not provide an interface to any other programming languages. For an additional $395, however, Microrim will sell another product called Program Interface that provides FORTRAN and Pascal programs with an interface to R:base 5000 databases.

Care must be taken when using resident programs of the SideKick variety with R:base 5000. The system used to review R:base for this article contains the Bellesoft Pop-Up Alarm Clock, which in one case caused a crash while R:base was running. The circumstances were unusual: the Pop-Up Alarm activated the computer's speaker at the same time that an R:base error message occurred (which also activates the speaker). The combination left the computer sounding like a continuous civil defense alert siren with no escape except to turn off the computer.

Two features of R:base 5000 promote data security and integrity. The first is passwords. Once passwords are defined for a database, users must enter the USER command and include a password before accessing the database.

R:base 5000 supports both owner and table passwords. If an owner password is present, only the owner can create or change the database structure. Also, only the owner can convert the data into ASCII format or into DIF or SYLK format for input into other programs. Table passwords limit access to specific tables and can define either read or modify permissions. Read permission allows the operator to examine but not change information in a table. Modify permission allows the operator to examine and change the table.

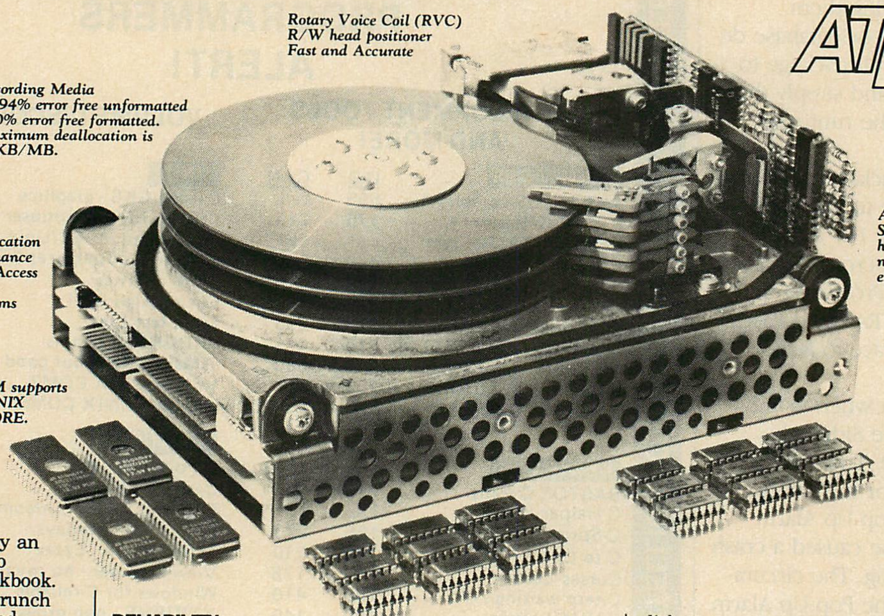
R:base 5000 also promotes data integrity by allowing the user to specify as many as 20 data-entry rules for each table. These rules, entered using the RULES command, prevent operators from entering invalid information. An

BEFORE YOU GET ON THE WRONG TRACK, READ THIS ABOUT THE ONLY ROM-BOOTABLE, PLUG-COMPATIBLE, XENIX SUPPORTING, HEAD-CRASH PROTECTED, HIGH-PERFORMANCE HARD DISK UPGRADE KIT FOR YOUR IBM PC-AT.

ATplus T.M.

ATplus drives are are manufactured for CORE by CONTROL DATA

Full one year warranty.
Service available worldwide.
(drive shown with cover removed)



THE BALANCING ACT OF 1985.

You didn't buy an IBM PC-AT just to balance your checkbook. You bought it to crunch lots of numbers and words, in the shortest possible time. A labor saver. A time saver. Hence, a money saver.

So, do your part for effective money management; hard disk storage is no place to be penny-wise and pound foolish.

TELL 'EM YOU NEED HIGH SPEED AND DATA PROTECTION.

These and other important features do add cost, but that makes a premium drive.

Anything that can be made, can be made cheaper, sell for less, offer lower performance, and probably die young.

Remember, usually you get what you pay for, and you ALWAYS get what you don't.

ALL HARD DISKS ARE NOT CREATED EQUAL.

There are vast differences in the speed and reliability of Winchester hard disks. Since the IBM PC-AT is an incredibly fast machine, a slow drive can make an AT run like an XT.

So, before you get stuck with a slow drive in your AT, save your boss two grand and buy an XT.

Or better yet, buy the AT and avoid any drive with Access Times over 40 milliseconds.

RELIABILITY: WHERE HAS ALL THE DATA GONE?

Now tell 'em the drive must have a data protection scheme. One that's easy to use and reliable.

Winchester heads read and write while "flying" a few microns above the data surface. If the heads contact the recording media, you risk a head crash, and significant or total data loss.

So, even a fast drive without data protection is virtually worthless. Frankly, we'd rather sleep at night.

BEWARE OF USER-DEPENDENT PROTECTION SCHEMES.

Some drives have a safe landing zone for the heads, but you need to call a separate program to send 'em there. If you don't call that program, and most folks won't, the heads in these drives ALWAYS land on data when powered down.

The slightest bump or vibration can move the heads, wiping out those data tracks. And the R/W heads can become contaminated, thus increasing the error rate, slowing down average access until the whole drive fails.

Consequently, those drives offer a very high risk of head crashes, a false sense of security, and little else.

What's your data worth? $200? $400? Specify AUTOMATIC data protection. ATplus has it. And it doesn't cost, it pays.

PEACE OF MIND.

Specify AUTOMATIC park and lock of the heads on power down.

This system provides unparalleled head crash protection, by sensing power loss to the drive, and retracting the heads to a dedicated landing zone before they can land on your data.

Since this is 100% automatic, user-dependent risk is eliminated.

OUR DRIVES HAVE ALL BEEN TO BOOT CAMP.

Avoid drives that CLAIM PC-AT compatibility but can't BOOT the AT. By the time you juggle the diskettes necessary to use one of those drives, the phrase "user-hostile" will have deep personal significance.

We believe that computers ought to serve people, not the other way around.

BEWARE OF THE BARGAIN BAND-SCHLEPPER.

Avoid drives with inexpensive Band-Stepper positioner technology. These were pretty good way back in 1980, considering that's all anyone had. But by today's standards, they're inaccurate and very mechanical.

They waste time looking for the right track to read or write. And they're worth no where near the price you'll pay for 'em --in more ways than one.

THE HIGH TECH SOLUTION.

Specify state-of-the-art ROTARY VOICE COIL (RVC) head positioner and CLOSED LOOP SERVO technologies.

This system uses a DEDICATED SERVO SURFACE (DSS) for continuous head location information.

The RVC and DSS use a large scale microprocessor to translate new track-seek commands into current that is applied directly to the RVC.

This moves and holds the R/W heads at the exact track intended.

Speed is dramatically increased since head movement is created electronically, with minimal mechanical interfacing.

BESIDES BEING THE SOLE SOURCE.

Specify names you know and can trust. Like Control Data Corporation, a world leader in hard disk technology. And CORE INTERNATIONAL, a growth company with over ten years in computer technology, design and manufacturing.

ATPLUS: IDEAL FOR THE SPECS MANIAC.

ATplus High Performance Hard Disk Upgrade kits, available in 20, 30, 40, 56, and 72 MB capacities.

They have ALL the above features and none of the shortcomings of "bargain" drives.

And your AT will accept any two ATplus drives, for up to 144MB of bootable, reliable, DOS and XENIX supporting high-performance on-line storage, all inside the box.

TELL YOUR DEALER...

That only ATplus Hard Disk Upgrade Kits from CORE will do.

If he or she doesn't have 'em, we'll gladly set them up as ATplus authorized dealers.

CORE INTERNATIONAL

542 S.E. 5th Avenue · Delray Beach, Florida · 305/276-3929

CIRCLE NO. 166 ON READER SERVICE CARD

error message that contains a maximum of 40 characters can be displayed whenever an operator enters a value that does not adhere to the rules.

For example, in the sample database, the STATE column in the AUTHOR file is a two-character field that is supposed to contain a valid abbreviation for one of the 50 United States. To ensure that only valid abbreviations are entered into this field, a new table was created and a rule was defined. The table (called STATES) consisted of a single column (called ABBREV) with 50 rows, each containing a state abbreviation. Defining the rule involves entering the following commands:

```
RULES
"Invalid State." state IN author EQ abbrev
  IN states
```

The second line says that the column STATE in the table AUTHOR must equal one of the elements in column ABBREV of table STATES. If an operator attempts to assign a different value to a state, the message "Invalid State." appears. Variations on this command can be used to prevent duplicate entries or to limit entries to a range of values.

R:base 5000 does not provide a mechanism for updating multiple tables simultaneously. For example, if a column called LASTNAME is included in two different tables, correcting an error in a last name involves two separate commands, one to update each table.

Data compression is not automatic in R:base 5000. When a user deletes a row from a table, the program does not compress the table to free disk space. The user must explicitly invoke a PACK or RELOAD command. Although this can be done at any time, Microrim recommends backing up the database first. A power glitch during this time could render an entire database unusable.

R:base 5000 sorting and indexing capabilities are powerful and easy to use. The SELECT command lets the user display information from a table. A SORTED BY clause can be added to specify the order in which items are displayed. For example, the list of authors might be sorted and displayed alphabetically by last names.

Additional sorting criteria also can be specified. In fact, each list can be sorted based on information in as many as ten columns, in either ascending or descending order. The same list of authors might be sorted by last name followed by first name.

R:base 5000 can produce key indexes to speed up database searches. For example, if the user constantly searches the AUTHOR file based on an author's last name, the searching process can be speeded up by having the last name column be a key column.

The BUILD KEY command is used to create a key index for a particular column. DELETE KEY deletes an index. R:base 5000 puts no restrictions on the number of key indexes per table. All columns can be key columns if desired, although only columns that contain many unique values really benefit from this designation. For example, if a column listing a person's sex (M or F) is a key column, searches based on that column might take even longer than before. Each key index requires approximately 12 bytes per key value. If information is added to tables, R:base 5000 automatically maintains the key indexes.

Reports can be created with R:base 5000 in two ways. Application EXPRESS can create simple reports that list specified columns from a particular table, in either row or column orientation.

The REPORTS command is used to design more complicated and useful reports. This command operates very much like the FORMS command does to create forms. Menus and prompts help the user along. First, he associates a particular table with a report. Although information from several tables can appear on a report, each report is based on a single table, with information from other tables read into the report using special look-up variables. Then the report can be edited to contain any text that is always to appear on it. The user defines the variables that will be used to produce data on it. These variables can be set to a constant, a column of a table, a global variable, or a value derived by performing an arithmetic operation on any two of these. A special operator, called SUM OF, is useful for computing totals of a column. After the table variables have been defined, the user locates them on the report, creating the format in which it will be printed. This process is just like locating values in a form.

Finally, the user marks the report to specify where and how often the various lines of the report will be printed. Four basic kinds of markings can be used: heading, detail, breakpoint, and footing. Headings are those lines that get printed only at the beginning of a report or at the top of a new page. A report-level heading could have the title and any other preliminary information. A page-level heading could contain column headings, the date, the time, and the page number. Detail lines make up

the bulk of the report, with the values pulled out of the tables. Breakpoint lines are used to print subheadings or subtotals. Each breakpoint is based on a particular column in the table or on a particular variable. Whenever the value of that column or variable changes, a new subheading or subtotal is printed. Page ejects also can be inserted before or after the breakpoint material. Each report can contain ten breakpoints. The footing contains information at the bottom of the report, such as grand totals or summary information.

Each page of a report can be 131 characters wide and 999 lines long. R:base 5000 permits the insertion of printer control codes to enable compressed printing, bold or underlined type, or other capabilities of the printer.

Few tools are provided to aid in crash recovery. The program has no explicit support for checkpoint/restart, backing out transactions, or audit trails. The most effective crash protection measure is frequent backups.

The current version of R:base 5000 does not support multiple users. Microrim plans to make a LAN version, R:base 6000, scheduled for release in the fourth quarter of 1985. This version will have multiuser capabilities and include record- and file-locking features.

Although the R:base 5000 documentation set is handsomely produced, with binder, slipcase, and typeset text, the contents do not match the quality of the rest of the product. Both the *User's Manual* and the *Reference Manual* have problems that make learning R:base 5000 more difficult than it should be.

The *User's Manual* is poorly organized. It has 16 chapters contained in three major sections called "Getting Started," "Managing a Database," and "Building an Application." The chapters do not progress logically and contain many forward and backward references. The differences between "Managing a Database" and "Building an Application" are indistinct so that it is unclear when to look in which section.

The table of contents and index only make matters worse. The main table of contents lists only the chapter titles, some of which are nebulous. More detailed listings precede each of the chapters, but the entire manual must be skimmed to find them.

The index is combined with a glossary, an innovative concept. However, the combination hides the fact that the index itself is small and incomplete. Several index entries refer to pages that refer to still other pages for more information. The second page numbers were not listed in the index.

The *User's Manual* is also littered with mistakes and omissions. In the tutorial chapters, some simple but essential steps are omitted and some table names are switched. Although this is not a major problem for experienced PC users, these errors could cause problems for beginners. In at least one instance, a mandatory keyword is omitted from the description of a command. Furthermore, the advanced tutorial covers mainly Application EXPRESS, a piece of the program that really does not require an extensive tutorial. Where tutorial help is really needed, such as in setting up complicated reports or building variable forms, it is not provided.

The *Reference Manual* is a better organized book. It provides detail about every command in the product, listed in alphabetical order by command name. This manual could have been made even better if it also contained usage examples for each command.

Unlike the printed documentation, the on-line help is quite good. Two different kinds of help are offered: textual

and prompting. Pressing the F10 key anywhere in the program displays a help menu. Users can select from the menu or type the name of a command for which they need help.

Product support is primarily handled through Microrim's listed telephone number. For an extra $150, however, users can subscribe to a software maintenance program, including use of Microrim's toll-free number, a monthly newsletter, and free product updates. For $7.95, Microrim offers *Microrim Applied*, a guide to existing applications, consultants on the Microrim product family, and training centers.

END-USER FACTORS

So far, the information in this article has applied primarily to developers—the persons who design databases and write applications that manipulate and report on the data. But the people who use a database product most are end users—those who enter new data or transactions into the system and who print out reports. What features does R:base 5000 provide for end users?

This product is easy to install and is not copy-protected. All the user must do to install the program is copy the disks into a directory, including the line FILES=20 in CONFIG.SYS. The *User's Manual* provides clear instructions.

Application EXPRESS makes menu design so easy that any application can contain an umbrella menu structure. This makes the end-user job easier, because all he needs to do is choose an option from a menu.

The program uses the PC's function keys for help and for a few other purposes. The cursor can be moved using the arrow keys, tab keys, and sometimes the Enter key and the user can exit or proceed to the next screen using the Esc, PgUp, and PgDn keys. For the most part, the use of these keys is fixed. (Other keys cannot be used instead.)

Graphics characters and color can be included in the design of screens. However, only foreground and background colors can be selected, not the colors of individual characters. Reverse video can be used for the cursor.

Programmer-written applications might not answer all the questions an end user needs to know. Many users will need to use the R:base commands themselves to obtain some of the information they need. In this respect, R:base 5000 is good. It uses English-like commands and provides prompting to guide a user through them. The command structure is similar to the IBM SQL product offered on mainframes.

Three R:base 5000 commands are particularly useful for ad hoc queries: SELECT, LIST, and TALLY. SELECT is the primary query command and displays entries from a table. Its English-like command structure is demonstrated in the example that follows:

SELECT title firstnam lastname FROM article SORTED BY lastname firstnam WHERE department EQ "Product Review"

This command displays a list of article titles and author names, sorted by author name, from a table called ARTICLE. It displays only the articles in the category "Product Review." When numeric values are displayed, another option can be added to display column totals.

COMPUTE and TALLY provide the user with other computational features. COMPUTE can list the average, count, maximum, minimum, sum, and number of rows for a column in a table. TALLY lists the unique values in a column and displays the number of times each value occurs. To use these commands, the user must know the names of tables and columns. To find out what those

names are, he can invoke the LIST command. Additionally, a prompt mode provides help while the user enters commands. Typing PROMPT followed by a command name displays some information about the command and shows the possible clauses that the command can accept. Then descriptions of each parameter appear, prompting the user to fill in the proper values.

These commands allow the display of almost any information from one table. If users need to display information from more than one table, they must use the relational commands to form new combined tables first, or they must write their own command files to display reports and forms.

When setting up a new database, many users want to read in files that were set up using some other scheme. In addition, some users would like to use spreadsheet products to process the information they store in their database. R:base 5000 provides a means of converting to and from a variety of other software programs.

This conversion can be performed using R:base commands or the FileGateway utility, which provides a menu-driven interface for transferring files. Both of these mechanisms allow conversion to and from DIF files (used by VisiCalc and others), SYLK files (Multiplan), ASCII files with fields in fixed locations, ASCII files with fields separated by delimiters, Lotus 1-2-3 worksheet files, dBASE II files, and pfs:file files.

Using FileGateway is the simplest way for the end user to convert files. This utility prompts for information along the way, making file transfer straightforward. When loading information from other formats, the user must specify an existing table in which to load the information. If FileGateway notices discrepancies between the structure of the table and the data being loaded, it puts the rows in question into an error file. The user can then examine the rows and change the information appropriately.

Converting to and from several other programs can be performed using R:base 5000 commands or the FileGateway utility.

The sample database used for this article was loaded using FileGateway. The information was supplied in both delimited ASCII and dBASE II format. FileGateway loaded the delimited ASCII versions without problems, but it halted part way through when attempting to load the dBASE II files. Examination of the error file showed nothing unusual about the data.

Because it is a fairly new product, R:base 5000 does not yet have many third-party applications. An order-entry application is included with the product. The application can be used as is, or it can be modified to suit the individual user. The code also can be used as an example of how to perform certain kinds of operations.

THE SAMPLE APPLICATION

R:base 5000 performance was measured by implementing the sample application designed by *PC Tech Journal* (see "Sample Application Specifications," August 1985, p. 48, or contact PCTECHline). This application is a subset of an editorial inventory system that a magazine could use to keep track of its authors, articles, and issue numbers.

The data for the application consists of three files. The AUTHOR file lists information (such as name, ad-

dress, telephone number, social security number, biography) for each author. The ARTICLE file lists information about each article (such as author, title, category, issue, date received, and payment). The ISSUE file lists information about each issue (such as its volume, number, deadline, and date).

The first step in the application design is to set up the structure of the tables that would hold the data. With the help of the specifications provided and Application EXPRESS, this process took less than five minutes.

Next, the information is loaded into the database. FileGateway encountered two problems during this operation. In the first, the operator's fingers were able to work so fast that the ARTICLE file was loaded into the AUTHOR table. Naturally, this produced a large error list. A quick examination of that list revealed the problem, which was then easily corrected.

The second problem occurred during the loading of telephone numbers as integers into the AUTHOR file. The ten-digit numbers exceeded the nine-digit size that R:base 5000 allows for integers. After a quick trip back to Application EXPRESS to change their data type, loading of the delimited ASCII files proceeded smoothly.

At first glance, Application EXPRESS looked as though it could handle the entire sample application with almost no programming. But in actual use, it could generate only the menus and possibly a couple of data input forms. The rest had to be done manually.

Even though Application EXPRESS cannot handle all the complicated operations needed for the sample application, it does offer an easy-to-use interface, and, upon exit, it generates a source file of R:base commands that can be modified. The application at least can be sketched out using Application EXPRESS; detail can be added by modifying the source output file.

Some operations are extremely easy to perform using R:base 5000. Other seemingly simple tasks require several lines of code.

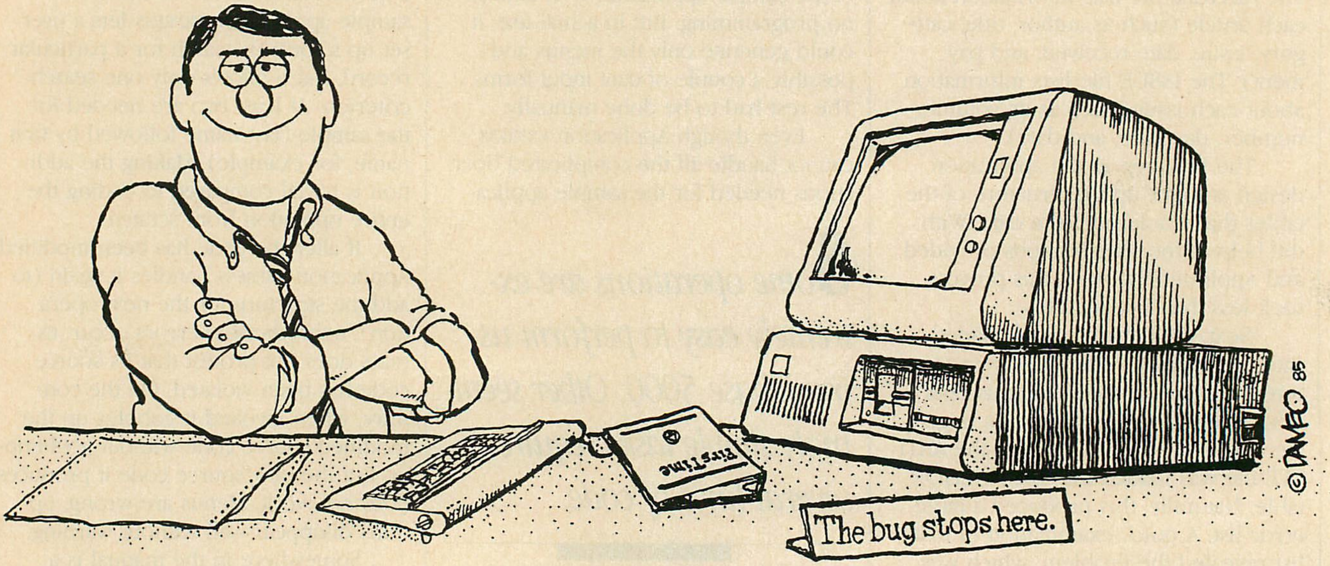
An important point to remember, however, is that once the file created by Application EXPRESS is modified, Application EXPRESS cannot be used again. Such a modification had to be made in the sample application. EXPRESS lets a user set up a form to search for a particular record, but it allows only one search criterion. At least two are needed for the sample (last name followed by first name, for example). Making the addition is trivial compared to coding the entire operation from scratch.

If after the code has been modified Application EXPRESS handles it again (to add the structure for the next operation), unpredictable errors occur. EXPRESS does not protest that its source code has been violated. On the contrary, when invoked it gobbles up the modified source code without a whimper, but the new source code it produces does not work. Menus are wrong, options disappear—all without warning.

Somewhere in the manual is a warning not to use Application EXPRESS after modifying the code manually, but, like most of the information in the manual, this warning is difficult to find or simply to come across. The manual and Application EXPRESS should be changed to add stern and obvious warnings about this situation.

The best solution is to build the application one piece at a time. Applica-

The First Idea-Processor For Programmers.



FirsTime™

Has features no other editor has.

- Fast program entry through single keystroke statement generators.
- Fast editing through syntax oriented cursor movements.
- Dramatically reduced debugging time through immediate syntax checking.
- Fast development through unique programmer oriented features.
- Automatic program formatter.

FirsTime is a True Syntax Directed Editor.

As the world's most advanced syntax-directed editor, FirsTime lets you work with ideas by taking care of the low-level syntax details of your program. For example, you can generate complete statement skeletons with one keystroke. Move the cursor from one procedure to the next with one keystroke. Type in code, and it's instantly formatted (you specify the rules). Type an error, and FirsTime warns you immediately. You can continue working if you wish. Later, you can use the search-for-error command to find the error and fix it.

FirsTime Has Thorough Error Checking.

FirsTime not only checks your syntax, but also semantics. FirsTime identifies:

- Undefined variables, types and constants.
- Assignment statements with type mismatches.
- Errors in include files and macro expansions.

FirsTime Lets You Work With Ideas.

The *Zoom command* gives you a top down view of your program logic.

The *View macro command* shows the expansions of a C macro while in the editor.

The *View include file command* instantly shows you the contents of an include file.

The *Transform command* allows you to change a statement to another similar statement, for example, a *FOR* to an equivalent *WHILE.*

The *Search for next error* command allows you to find errors throughout your program.

The *Cursor movement commands* let you traverse your program by logical elements, not just characters.

FirsTime Works With Existing Files.

FirsTime works with standard ASCII files, so you can edit any existing source files.

FirsTime for Turbo Pascal	$ 74.95
FirsTime for dBase III	$125.00
FirsTime for MS–Pascal	$245.00
FirsTime for C	$295.00

To Order Call: (201) 741-8188 **or write:**

SPRUCE TECHNOLOGY CORPORATION

P.O. Box 7948
Shrewsbury, NJ 07701

In Germany, Austria and Switzerland contact:
Markt & Technik Software Verlag
Munchen, W. Germany
(089) 4613-0

TABLE 1: *Benchmark Results*

BENCHMARK TASK	TIME (secs)
Add 900 records to an empty database	105
Index database on two fields (7 bytes)	75
Document and tally codes from one field	8
Mass change of one field	12
Extract selected records to create a text file	3

R:base 5000 performance is especially impressive when adding the 900 records to the AUTHOR data file. Counting the occurrences of each state code is also fast because one command, TALLY, does the job.

tion EXPRESS can be used to come as close as possible to the desired operation of each piece; the code can be tweaked manually to add the necessary extras to the particular database.

Coding the application to provide the features demanded by the *PC Tech Journal* application took approximately one 40-hour work week, with much of that time spent in learning how to use the product. The documentation's poor organization and lack of examples slowed the process. The examples provided were very helpful. After gaining experience with R:base 5000, a comparable project could probably be accomplished in about half the time.

The capabilities of R:base 5000, however, proved to be more than adequate for the task at hand. Whatever the sample application demanded, R:base 5000 provided it—often in several different ways. Application EXPRESS was used to set up the general menu structure and to generate code for the simpler parts of the application. EXPRESS can handle adding new information to a single table, modifying a single table, or deleting a row from a table. Implementing the sample data entry screen, shown in photo 2, as well as designing the reports, requires custom programming. The code fragment required to accept data from the sample data entry screen is shown in listing 1.

Reports are set up easily using the REPORTS command. Mailing lists are especially easy because R:base 5000 includes a special command file to produce them. Debugging also is an easy task. The SET ECHO = ON command displays the commands as they execute, including the command inputs. This is almost like single-stepping through code with a debugger.

However, the program does have some frustrating limitations, as mentioned earlier. While processing a command file, R:base reads the entire WHILE loop into a memory buffer, which is limited to 4,000 bytes, regardless of the amount of memory in the computer. Considering that menu selections are usually implemented as WHILE loops, this limits the amount of code for a menu to 4,000 characters.

Another limitation also concerns menus. R:base 5000 permits only nine selections in any menu, even those designed outside Application EXPRESS. In addition, R:base puts an arbitrary but unstated limit on the amount of text the menu can contain. Exceeding the limit results in an error message.

The program contains some bugs. One surfaced while selecting the final versions of the forms for the sample application. Several unused forms were left in the database during development. Later, they were edited and deleted. During this process, an internal error occurred, terminating the forms-definition phase of the program. This error persisted until R:base was exited.

Another bug involved comments in R:base code. The format for a comment is to surround the comment with parentheses and precede it with an asterisk. During the sample application, these characters were used temporarily to prevent some of the code from running during the test phases of development. However, when the comment characters were placed around the command SET ERROR MESSAGE OFF, R:base interpreted the comment as an improperly constructed command.

The sample application is required to perform several benchmarks and ad hoc queries. Table 1 lists times required to perform these operations. R:base 5000's performance is spectacular in loading the large data file and quite favorable in the other operations. R:base 5000 was written in Pascal, compiled using the Microsoft Pascal compiler, and linked with Plink86.

Some operations are extremely easy to perform using R:base 5000. Other seemingly simple tasks require several lines of code. One of the benchmarks required by the sample application calls for listing each unique state code in alphabetical order and counting

the number of occurrences of each one, requiring the following command:

```
TALLY state FROM author
```

Similarly, finding all the authors who live in California, sorting the list by zip code, and writing the list to an external file in delimited ASCII format involves just the following command:

```
UNLOAD data FOR author USING all AS
   ASCII SORTED BY zip
```

However, some simple-sounding queries require several commands and the use of variables to store temporary results. Counting the number of editorial, listing, and total booked pages in an issue (volume 3, number 2) required the following commands:

```
COMPUTE edpg AS SUM ed-pages FROM
   article WHERE volume EQ 3 AND
   number EQ 2
COMPUTE 1spg AS SUM 1s-pages FROM
   article WHERE volume EQ 3 AND
   number EQ 2
SET VAR totpg TO .edpg + .1spg
SHOW VAR edpg
SHOW VAR 1spg
SHOW VAR totpg
```

In general, whenever an operation calls for totaling values from multiple columns or basing a selection on criteria listed in another table, multiple commands and variables are required. Listing the amount paid for articles in a given issue involves a single command:

```
COMPUTE SUM payment FROM article
   WHERE volname EQ 3 AND number
   EQ 2
```

But computing the payments, bonuses, and total payment requires a separate COMPUTE command to calculate the payment and bonus and to store the values in variables. Two more commands are required to add the payment and bonus totals and print the result. Fortunately, these commands are straightforward. Of the queries required by the sample application, only determining how much is paid per printed page for editorial material in an issue and determining which articles are received after the deadline for an issue involve more than rudimentary programming knowledge.

LIVING UP TO EXPECTATIONS

Microrim sets high expectations for R:base 5000. Indeed, the product meets many of its advertised claims. It is probably as powerful a database manager as any in the PC marketplace. The Application EXPRESS and FileGateway utilities make getting started easy.

Microrim's claims could lead the nonprogrammer to the mistaken belief that Application EXPRESS will permit him to build complicated database applications. However, to create real applications worthy of a $700 program, programming knowledge is necessary, just as with any other database manager. Application EXPRESS is nonetheless a valuable tool, allowing nonprogrammers to set up the database structure, design menus, perform simple operations, such as entering or changing information in a table, and design reports.

Learning the R:base programming language is relatively easy. Experienced BASIC, FORTRAN, Pascal, or C programmers will feel at home immediately. Nonprogrammers will find that learning the R:base 5000 language is comparable to learning BASIC. The lack of programming examples in the documentation, however, makes the job a little more difficult than it needs to be.

The greatest drawback of R:base 5000 is its documentation. The manuals need work to bring them up to the level of the rest of the product. A better ta-

ble of contents and index, along with more examples and a thorough proofreading could go a long way in turning a good product into a great one.

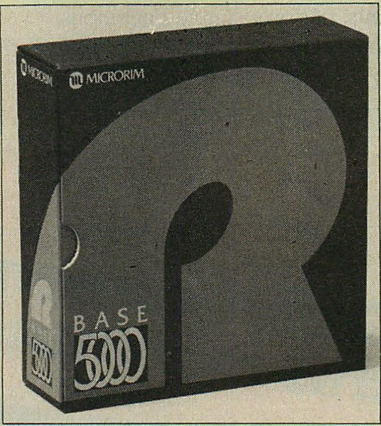
R:base 5000 is powerful enough to be used in a business environment. It gives users a head start in their applications development, although perhaps not quite as much as Microrim would like potential buyers to believe. The commands are straightforward and understandable, and the product has exemplary prompting and help information. It can be recommended for anyone who needs a stand-alone, top-of-the line database manager.

Ted Forgeron is a microcomputer software consultant. Steven Armbrust is a freelance technical writer. They work primarily in the "Silicon Forest" west of Portland, Oregon.

R:BASE 5000 OVERVIEW

R: BASE 5000, version 1.0
Microrim, 3380 146th Place SE, Bellevue, WA 98007. 206/641-6619.

Product type. Relational database management system for programmers.
Software environment. PC-DOS 2.x and higher. Multiuser version to be announced in fourth quarter of 1985.
Hardware environment. IBM PC, XT, AT, and compatibles with a minimum of 256KB of memory. Also supported are additional RAM, RAM disk, and hard-disk drive.
User interface. Program is command language and menu-driven. It uses macros/procedures. The command language, data definition, and data manipulation are English-like.
File limitations. 1,500 characters per field; 1,530 characters per record; 400 fields per record; unlimited number of records per file; The basic architecture is relational.
Access to system facilities. From within the program the user has the ability to display a directory of files on a disk, delete, copy, and type files, and change a subdirectory.
File modification facilities. The program can merge two or more files into one and can split one file into two or more files. It can update a file with data from another file and update multiple files simultaneously. The user can add fields without experiencing loss of data in file.
Help facilities. Online help and a written tutorial are provided.
File design. The product uses a screen-painting method of entry-screen design, allowing custom data-entry screens; users may create multiple data-entry screens for a single file; entry screens may be more than one screen in length. The program supports derived fields, with the information derived from calculations, another file, or a user-supplied list or file of acceptable values. Other field attributes allow for view-only, required, and double-entry fields that force the operator to re-enter information to verify accuracy.



Data entry. The program provides range checking functions and allows the user to supply standard entry values during entry; it provides facilities for batch data entry.
Query and sorting. Search facilities allow for selection and logical operators. Sorts may be performed in ascending and descending order. The product supports multiple sort operations on up to ten fields and multiple indexes on up to 400 fields. Query and ordering specifications may be saved for repeated use. Automatic updating of indexes is offered.
Reporting. The program produces label reports, which may produce two-or-more across labels. Report formats may be edited and may contain information from two or more files. The program produces summary reports. Reports are created using a screen-painting method. They may include totals and subtotals, control breaks for pagination, and calculated results using four-function math. Final reports may include headings, footers, and pagination; they may be directed to the screen, printer, disk file, or import/export file format. Print enhancements, such as bold or underscore, may be used.

Security. The program provides password-protected access to the program itself, multiple levels of password protection, and data encryption.
Utilities. File maintenance and back-up utilities are provided.
Applications development facilities. Customization may be performed through the use of macros/procedures and custom menu generation. The program provides a fully programmable procedural language. It can generate turn-key applications. A runtime compiler or module is available for applications.
Data compatibility. The program reads DIF, Lotus, DBF (dBASE), SYLK (Multiplan), and comma-delimited ASCII file formats.
Printers supported. More than 25 major printers are supported.
Special features. A compiler for binary encoding command procedures is provided. Program enhancements available separately include Extended Report Writer (XRW), which expands report-writing capabilities; Program Interface, which provides a library of routines to allow Pascal and FORTRAN applications to run with R:base; CLOUT, a natural language query capability for analysis of data that provides access to data in other file formats.
Delivery. Began in April 1985.
Price. Retail: $700; upgrade for R:base 4000 to R:base 5000: $245; Extended Report Writer: $150; Program Interface: $395; CLOUT: $249; runtime version: $450.
Support. The product includes on-screen tutorials, sample applications, telephone support, and a demonstration diskette. A Software Maintenance Plan is available for a cost of $150 per year; updates are included in support. Back-ups are permitted for personal use.

This material is taken from Data Decisions Microcomputers, *a three volume monthly updated reference service that evaluates microcomputer systems, software, and peripheral products. For additional information contact Data Decisions, 20 Brace Road, Cherry Hill, NJ 08034. 609/429-7100.*

LISTING 1: SCREEN.APP

```
*(Use Sample Data Entry Screen)

IF pick1 EQ 5 THEN
  SET VAR more TEXT; SET VAR more TO "Y"
  WHILE more EQ Y THEN
    CLEAR ALL VARIABLES
    SET VAR more TEXT; SET VAR more TO "Y"
    NEWPAGE
    DRAW invent WITH ALL AT 2
    WRITE "Fill in the title to use for search," AT 1 1
    WRITE " then press [Enter]." AT 1 37
    WRITE "Press [ESC] to exit." AT 2 1
    EDIT VAR titvar RETURN ESC ENTER
    IF #RETURN EQ ESC THEN
      FILLIN more +
        USING "Search for a new article? (Y/N)" AT 24 10
    ENDIF
    IF #RETURN EQ ENTER THEN
      SET POINTER #3 ptr3 FOR article WHERE title = .titvar
      IF ptr3 EQ 0 THEN
        SET VAR frstvar TO firstnam IN #3
        SET VAR lastvar TO lastname IN #3
        SET VAR cofvar TO co-first IN #3
        SET VAR colvar TO co-last IN #3
        SET VAR volvar TO volume IN #3
        SET VAR numvar TO number IN #3
        SET VAR catvar TO category IN #3
        SET VAR deptvar TO dept IN #3
        SET VAR rcvvar TO date-rcv IN #3
        SET VAR edpgvar TO ed-pages IN #3
        SET VAR lspgvar TO ls-pages IN #3
        SET VAR payvar TO payment IN #3
        SET VAR bonvar TO bonus IN #3
        SET POINTER #2 ptr2 FOR issue +
          WHERE volume = .volvar AND number = .numvar
        IF ptr2 EQ 0 THEN
          SET VAR monvar TO month IN #2
          SET VAR yearvar TO year IN #2
          SET VAR deadvar TO deadline IN #2
        ENDIF
        SET POINTER #2 ptr2 FOR author +
          WHERE firstnam EQ .frstvar AND lastname EQ .lastvar
        IF ptr2 EQ 0 THEN
          SET VAR phonvar TO hm-phone IN #2
        ENDIF
        SET VAR pgtotvar TO .edpgvar + .lspgvar
        SET VAR pytotvar TO .payvar + .bonvar
        DRAW invent WITH ALL AT 2
        WRITE "Edit fields, press [PgUp] to save changes." +
        AT 1 1
        WRITE "Press [ESC] to exit." AT 2 1
        EDIT VAR titvar frstvar lastvar cofvar colvar volvar +
        numvar catvar deptvar rcvvar edpgvar lspgvar payvar +
        bonvar RETURN ESC PGUP
        IF #RETURN EQ ESC THEN
          FILLIN more +
            USING "Search for a new article? (Y/N)" AT 24 10
        ENDIF
        IF #RETURN EQ PGUP THEN
          CHANGE title TO .titvar IN #3
          CHANGE firstnam TO .frstvar IN #3
          CHANGE lastname TO .lastvar IN #3
          CHANGE co-first TO .cofvar IN #3
          CHANGE co-last TO .colvar IN #3
          CHANGE volume TO .volvar IN #3
          CHANGE number TO .numvar IN #3
          CHANGE category TO .catvar IN #3
          CHANGE dept TO .deptvar IN #3
          CHANGE date-rcv TO .rcvvar IN #3
          CHANGE ed-pages TO .edpgvar IN #3
          CHANGE ls-pages TO .lspgvar IN #3
          CHANGE payment TO .payvar IN #3
          CHANGE bonus TO .bonvar IN #3
        ENDIF
      ENDIF
    ENDIF
  ENDWHILE
  CLEAR ALL VARIABLES
  GOTO startapp
ENDIF
```

IT'S BETTER TO HAVE MORE POWER THAN YOU NEED THAN NOT ENOUGH.

There's a funny thing about database users. A lot of people buy a program that has all the power they need to get started. And not a whole lot more.

Well, that makes perfect sense. If you plan to stay in the same place all your life. But as most users get the feel of things, they want to push themselves farther and farther. That's when they wish they'd bought a program that had some extra muscle. And if you look at this chart, you'll see that's the whole idea behind R:base™ 5000.

DBMS POWER
R:base 5000 vs. dBASE III

	R:base 5000	dBASE III
Comprehensive procedural language	Yes	Yes
Built-in application compiler	Yes	No
Choice of row or column report format	Yes	No
Accessible tables per report	40	10
Number of relational operators	6	2
Password security	Yes	No
User-definable data entry rules	Yes	No
Automatic key index maintenance	Yes	No
Data dictionary	Yes	No

THE KIND OF POWER THAT'S EASY TO HANDLE.

We made R:base 5000 so easy to use that a complete novice can develop applications with it right off the bat. But that's just the beginning; we've built in a whole arsenal of high-powered development tools that the most intense programmers can't outgrow.

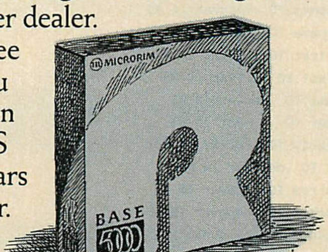
To give you a better idea what this extra power can do, just take a look at one feature. With R:base 5000, you get a relational operator called SUBTRACT, which dBASE III doesn't offer. By using this command, you could take all the names in one file (such as your customer list) and subtract them from any other file (like the Fortune 500). The result? An instant list of new business opportunities. And this is only one way to use one of the extra capabilities on this chart; there are literally hundreds more.

SEE FOR YOURSELF: 1-800-547-4000.

If you believe in common sense as much as we do, you won't take our word for it; you'll get your hands on a copy and see for yourself. And that's just what we'd like you to do. For only $9.95 (plus shipping) we'll send you a mini-version of the product that lets you build real applications.

Just call 1-800-547-4000 and ask for Dept. 817 . From Oregon, or outside the U.S., call 1-503-684-3000, Dept. 817 . We'll send your copy right out. If you'd like to see R:base 5000 today, head straight for a leading store or computer dealer.

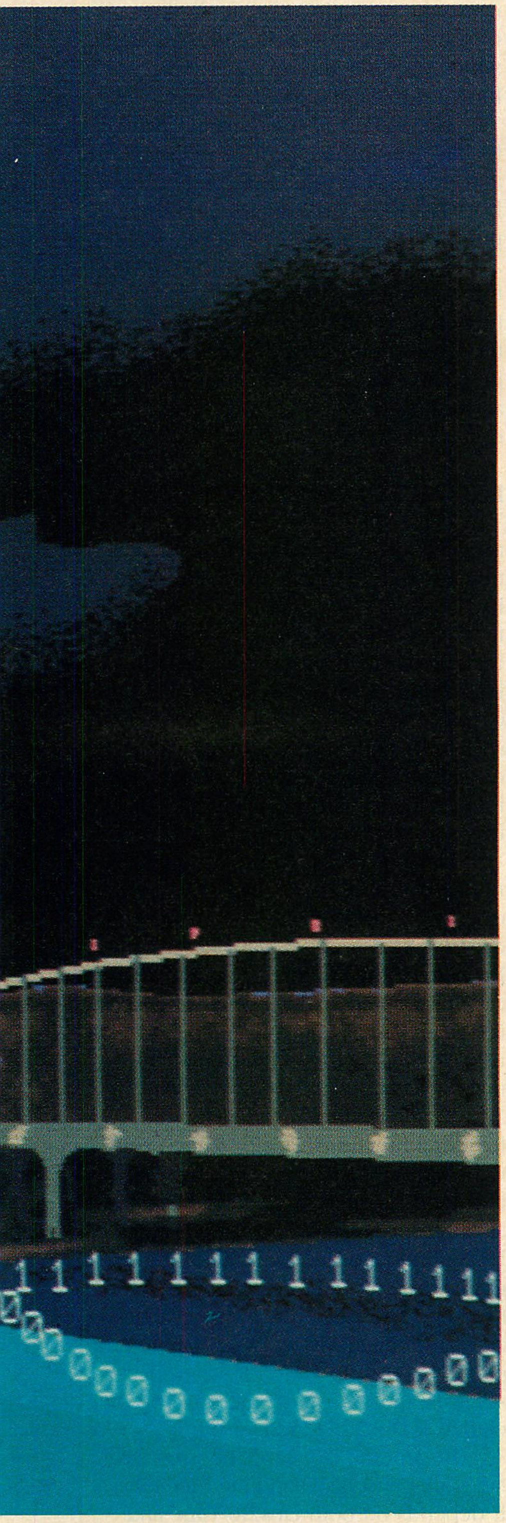
And see how far you can go when your DBMS has all its oars in the water.



R:BASE 5000
FROM MICRORIM
IT ALL COMES DOWN TO COMMON SENSE.

RUNTIME VERSION NOW AVAILABLE.

dBASE III is a trademark of Ashton-Tate.



Improving Turbo's I/O

A library of routines gives Turbo Pascal I/O capabilities at least as powerful as those in BASIC.

COLE BRECHEEN

Turbo Pascal, from Borland International, is taking the Pascal programming world by storm. Priced at $69.95, it is more powerful in many ways than Pascal compilers listed in the $200 to $500 price range. Like all programming languages, however, it will do little on the day it is taken out of the box. Even with a compiler as fast and full-featured as Turbo Pascal, a long period of tool building must be endured before even the first line of a sophisticated, commercial-quality program can be written.

The most important and most difficult stage of this process is likely to be the development of all the input/output facilities the program will need. Although Turbo Pascal 2.0 includes a more generous and more carefully designed set of intrinsic I/O facilities than most Pascal compilers, its I/O still suffers from two notable deficiencies. Either of these deficiencies may be critical to an application's success.

COMPUTER GRAPHIC • GARY ZAMCHICK / LEXICA

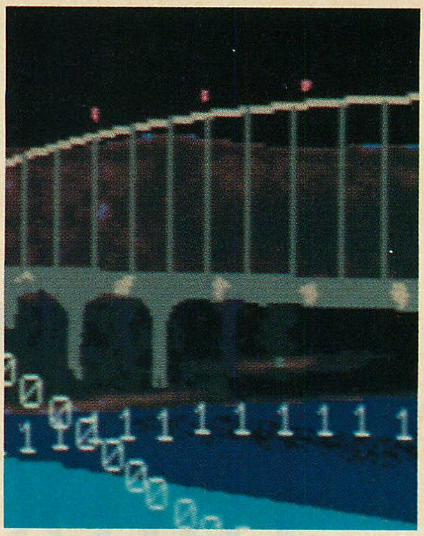
First, neither version 2.0 nor the previous issues of Turbo supports any of the more powerful file-handling and I/O functions of DOS 2.x and 3.x. Turbo's I/O cannot be used with directory path names or files outside the current directory and cannot be redirected from the DOS command line.

Second, Turbo's screen handling is terminal-dependent. Both Turbo's PC-DOS and its MS-DOS generic versions generate code that must be tediously installed each time the compiler is run with a different console or computer. Installation requires direct modification of a .COM file with a terminal-installation utility that is not included in the standard Turbo Pascal package. Furthermore, installed programs generated by the DOS version of the compiler will not necessarily cooperate with multitasking and networking extensions to the operating system.

This article presents a library of I/O routines that cures both of these deficiencies. It explains why and how to use the library and deals with the problems of handling path names, tree-structured directories, and I/O redirection. Source code for the library's routines is provided on PCTECHline.

The most important reason to use this library is that it lets Turbo programs operate in the tree-structured directory environment of DOS 2.x and 3.x. As hard disks and high-capacity floppy disks become more common, it is increasingly necessary for all DOS programs to work with path names. A program that cannot accept path names cannot manipulate files outside the current directory, which forces the user to change the current directory frequently to use the program. Even if the program can change the current directory, it cannot work with files from different directories at the same time.

There is no halfhearted way to make a Turbo program accept path names. Turbo uses *file control blocks* (FCBs) for all file I/O; FCBs are the file I/O mechanism used by DOS 1.x. A path name cannot be assigned to an FCB, and it is impossible to bridge the gap between FCBs and file handles, the I/O mechanism introduced in DOS 2.0. Not only will Turbo's **Read**, **Write**, **BlockRead**, and **BlockWrite** procedures not perform handle-oriented I/O, but its **Reset**, **Rewrite**, and **Close** procedures will not return, create, or close file handles. In order to use file handles with Turbo, the programmer will need a complete complement of handle-oriented file I/O routines, such as the **Dos2io** library that is presented here.



The only significant DOS 2.x functions that are not supported in the library are the extended memory and process-management routines, functions 48H through 4DH. These functions (in particular, the EXEC function, 4BH) would add greatly to the **Dos2io** library, but there is no obvious way of making them work from within a Turbo Pascal program. The extreme paucity of the IBM/Microsoft documentation for these functions makes it difficult to understand why they will not work in Turbo, but an educated guess can be made. According to Appendix E of the PC-DOS 2.0 user's manual ("DOS Control Blocks and Work Areas"), the following is true whenever the operating system loads a COM program:

> All four segment registers contain the segment address of the initial allocation block, that starts with the Program Segment Prefix control block. All of user memory is allocated to the program. If the program wishes to invoke another program through the EXEC function call, it must first free some memory through the **Setblock** (hex 4A) function call, to provide space for the program being invoked.

The **Setblock** function requires that, upon entry, the ES register must contain the segment of an allocated memory block. When the **Setblock** function is given the address in one of the segment registers at the time a Turbo program is loaded, however, it returns the error message, "Memory Control Blocks destroyed."

The PC-DOS 2.0 user's manual contains no hints about what a "Memory Control Block" might be, how it works, or whether a "Memory Control Block" and an "allocation block" might be the same thing. Chances are, however, that a Memory Control Block is a uniquely formatted area of memory created by the 48H (allocate memory) function. **Setblock** probably does not work with Turbo Pascal because Turbo never invokes the allocate memory function to begin with. The problem cannot be solved after a program is loaded, because the operating system allocates all of user memory to a .COM program, which prevents the allocate memory function from operating.

Hence, Turbo users are probably locked out of the extended memory and process-management routines until Borland does whatever is necessary to put a valid Memory Control Block at the address indicated by the segment registers when a program loads. Borland's technical support representatives have tentatively confirmed this analysis of the problem, but they have no suggestions about how it might be solved with the current version of the compiler.

LIBRARY ORGANIZATION

The Turbo I/O library presented in this article is divided into two main sections: **Dos2io** and **AnsiStuf** (see the PCTECHline for source code). **Dos2io** is further subdivided into three separate include-file modules to encourage selective use of its facilities. **Dos2io-1.inc** contains the most basic routines. It should be sufficient for programs that do simple screen output and keyboard input or that perform all file handling through I/O redirection.

Dos2io-2.inc provides all the basic file-handling functions. Most file-oriented programs will need to use it. It contains analogs of Pascal's **Reset** and **Rewrite** as the **OpenFile** and **CreateFile** commands; the analogs store a file handle in a normal integer. This integer is used for I/O to the file with the routines in **Dos2io-1.inc**.

In addition to the basic file-handling functions, **Dos2io-2.inc** provides routines that create, delete, and change directories; set and retrieve file attributes; retrieve the contents of the DOS command line at the time the program was executed; delete and rename files; and set and retrieve the system date and default drive. **Dos2io-2.inc** depends on **Dos2io-1.inc** for a number of services; thus, **Dos2io-1.inc** must be included if **Dos2io-2.inc** is used.

In general, **Dos2io-3.inc** provides fancy file- and disk-handling functions. It depends on both **Dos2io-1.inc** and **Dos2io-2.inc** and requires the inclusion of these two files. **Dos2io-3.inc** solves the ubiquitous problem of adding data to a file without recopying the data already there. It provides simple access to disk directories and subdirectories,

"An important new metaphor from the author of the best book on user interface design."
**—Alan Kay,
Apple Fellow, Apple Computer**

SNEAK PREVIEW
of a powerful
NEW SOFTWARE METAPHOR
WANTED: People with Imagination, an IBM PC, and $59.95.

by Paul Heckel President, QuickView Systems and author, *Elements of Friendly Software Design*

Rarely does a software product introduce a new conceptual metaphor. VisiCalc introduced the electronic spreadsheet; Thinktank, the electronic outliner; and now *Zoomracks*, the electronic rack. Let me tell you what electronic racks are, why I think they are important, and how you can get to try them risk-free at a savings now and maybe help shape their final form to your liking.

New Metaphor:
Originally designed to keep track of lists, names and addresses, appointments, notes, and other information on portable computers, electronic racks provide a simple, consistent and rich organizational metaphor for data base, text, and other applications.

Zoomracks starts with something familiar: racks—like those filled with time cards next to time clocks in factories. You can see the first line of each card, and take out a card to look at it in detail. You expect the cards in a rack to be in order, several racks to be next to each other; and to be able to move cards from one rack to another.

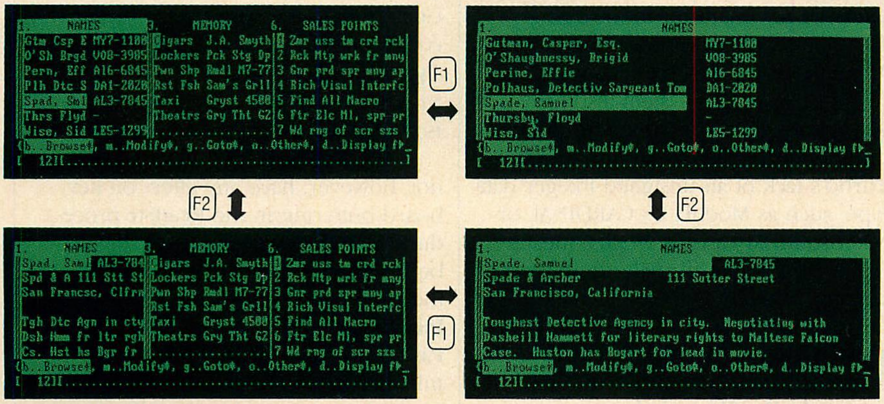
You might put names and addresses in one, appointments in a second, notes in a third, sales orders in a fourth, memos in a fifth, and archived appointments or notes (moved or copied from the second or third rack) in a sixth rack. To do something with Zoomracks, first ask yourself: "How could I do it with cards in racks?"

Windows illuminate like a flashlight in a dark room
Racks are displayed with Smart Zooms. While windows sacrifice the big picture to let you see the detail, Smart Zooms squeeze out the detail to always show you a recognizable big picture—whether a long shot of several racks, a closeup of one rack, or an extreme closeup of a single card.

One time offer
If you like to stretch new products and influence their final form, we want your feedback. We are making a one time offer of a Sneak Preview Edition of Zoomracks at an affordable price so you can try it and give us your feedback in time to make a difference before we officially introduce Zoomracks in November.

ZOOMRACKS SPECIFICATIONS:
- Copy and move fields, cards, and text into different fields, cards and racks.
- Define and change card templates.
- ASCII MS/DOS file format for conversion to other data formats.
- Utilities to convert DBASE II files.
- Macros.
- Simple Wordstar-like editor.
- Easy to learn and easy to use for both occasional and frequent users.
- Display sizes: 6 x 25 to 25 x 80.
- 8 racks on screen, in memory; 30 fields/card; 80 characters/line; 250 lines/field, 20,000 cards/rack.
- Runs on 256K IBM PC.

The Wide key (function f1) toggles between displaying the working racks (left two screens) and the current rack full (right two screens). Smart Zooms compress out detail to keep the big picture.



The Yank key (function f2) toggles between displaying the first lines of cards in racks (top two screens), and the current card (bottom two screens). In these pictures Zoomracks is using a 10 by 60 screen.

Before developing Zoomracks, Paul Heckel studied what made VisiCalc and other software powerful, useful, easy to use, and successful. He crystallized his thoughts in a book. This is what people are saying about this book, *The Elements of Friendly Software Design:*

"It is the first computing book I've ever read nonstop from cover to cover . . . one of the few books I've read on any topic that actually delivers what it promises . . ."
—Dave Bunnell, PC World Publisher

"Entertaining and instructional . . . will affect the way I program from now on."
—David Clark, Byte Magazine

"Informative, useful and entertaining. Will help improve your communications skills in any medium . . ."
—Robert Burton, President, Rolodex

The Elements of Friendly Software Design is available at your local bookstore for $8.95 or by calling 800-443-0100 EXT 341. You can also order by writing QUICKVIEW SYSTEMS, 146 Main St., Suite 404, Los Altos, CA 94022. Add an additional $2.50 for postage and handling. Payment must accompany order.

What you get for $59.95
- The Sneak Preview edition of Zoomracks;
- A free upgrade to the first run edition of Zoomracks;
- An acknowledgement in the user manual *if* you are the *first* to suggest an improvement we use;
- A six-month unconditional moneyback guarantee.

To order the Sneak Preview Edition
$59.95 copy protected
$79.95 copy unprotected

CALL
800-443-0100 EXT 341
OR
WRITE
QUICKVIEW SYSTEMS
146 Main Street,
Suite 404
Los Altos, CA 94022

CIRCLE NO. 135 ON READER SERVICE CARD

allowing directories to be listed in any format, using all the standard DOS wildcards from within a Turbo Pascal program. It lets the user set and retrieve a file's date and time and lets him read a disk's volume label. In addition, it includes convenient procedures for copying files and for determining the number of free bytes on a disk.

Briefly, **AnsiStuf** provides screen-handling functions (cursor addressing, screen clearing, etc.) that are transportable at the object-code level to any DOS 2.x/3.x computer. No terminal installation is required. **AnsiStuf** replaces and extends all the intrinsic Turbo screen-handling routines. Its routines allow redefinition of keys; provide control over graphics modes and screen colors; and allow text to be written in reverse video, high intensity, blinking, and underscored (if the display adapter supports it) modes.

HOW DOS2IO WORKS

Dos2io centers around the **msdos** procedure mentioned in the Turbo Pascal *Reference Manual*. The basic theory of the **msdos** procedure is that the parameter variables correspond to the 8086 family of registers. The value in each variable is moved into the corresponding 8086 register and an interrupt 21H is performed. This invokes one of the DOS service routines—which routine is invoked depends on the current value in the AH register. **Msdos** then restores the original values of registers that do not change during the interrupt and moves all of the register values back into the corresponding variables.

As currently implemented, the **msdos** procedure is enormously intolerant of error; the compiler cannot catch mistakes in the way **msdos** is used, and mistakes will usually crash the machine. The system access routines in **Dos2io** guard against such misbehavior with an extensive system of error trapping.

Another problem is the documentation for using the **msdos** procedure. The deficiencies are not solely errors of omission. At the time I bought the compiler, all of Borland's attempts to document the routine—even the on-disk corrections to the original documentation—were wrong about the structure of the record **msdos** requires. The record must have 20 bytes, not 18, and the register variables must be listed in the order set out in **Dos2io-1.inc**.

The structure of the simpler **Dos2io** routines is as follows. **Rgstr a.h** (**rgstr** is the name that most of the routines use for the variable passed to **msdos**) is set to the hexadecimal number of the **msdos** service routine to be called. Input data are then converted into the format the operating system expects. This involves some unusual twists, some of which are explained below.

The PC-DOS 2.0 user's manual contains no hints about what a "Memory Control Block" might be, how it works, or whether a "Memory Control Block" and an "allocation block" might be the same thing.

Msdos is called, and if all the right conditions are met, the operating system does its work. Otherwise, the operating system signals that there has been an error, and one of two things happens: either the procedure **PrintMessage** reports the error type and halts the program or the error type is passed back to the calling procedure through the function name, and the calling procedure assumes the burden of responding appropriately. If there has been no error, **Dos2io** retrieves the service routine's output from the proper **rgstr** fields and converts the data into the format used in Turbo Pascal.

Data format conversion. Converting data from the format used by Turbo Pascal to the formats expected by the operating system can be tricky. For example, the operating system does not use the type of character string used by most implementations of Pascal, including Turbo. The operating system uses "ASCIIZ" strings, which are essentially Pascal strings with no length byte in the zero position but with a byte of zeros after the last character in the string. **Dos2io** includes the procedure **MakeAsciiZ** to handle the conversion.

The operating system also has sense enough not to be confused by numbers between 32,767 and 65,536. Turbo's lack of an unsigned integer data type, such as Modula-2's CARDINAL or Microsoft/IBM Pascal's WORD, means that Turbo thinks all such numbers are negative. **Dos2io** is forced to handle these numbers as reals until just before passing them to **msdos**; negative results from **msdos** must then be converted into reals. **Dos2io** includes two procedures—**WordToReal** and **RealToWord**—to handle the conversions.

Turbo differs from the operating system even more in the handling of numbers larger than 65,536. The operating system uses the same segment/offset scheme that is used by the 8086 family of processors to represent large physical addresses in memory. Turbo, of course, uses reals, which bear no resemblance to segment/offset values. **Dos2io** includes two procedures—**SegmentedToReal** and **RealToSegmented**—to handle these conversions.

Keyboard buffering. A close examination of **Dos2io-1.inc** will reveal that it contains two apparently unrelated attempts to control keyboard buffering. At the top of both **Dos2io-1.inc** and **Dos2io-2.inc** are the compiler directives {$U-} and {$C-} with the comment, "Enables keyboard buffering." However, **Dos2io-1.inc** also declares a global boolean variable, **TypeAheadLegal**; its purpose is to let programs change the value of **TypeAheadLegal** and thereby control whether **GetKey** and **GetExtendedKey** will clear the keyboard buffer before looking for another character.

Keyboard buffering must be handled in two steps because Turbo changes the vector of the keyboard interrupt service routine in order to allow Turbo's own, customized code to filter input from the keyboard. The compiler directives {$U-} and {$C-} determine whether Turbo's keyboard interrupt service routine will maintain a buffer. (This feature is not documented in the compiler manual; it was discovered by calling Borland's technical service.)

The technique that Turbo Pascal uses affects the ability of routines that get their input through the operating system to use a buffer, because the operating system's keyboard-input services determine the location of the keyboard interrupt service routine by looking at the vector table Turbo has modified. Therefore, if neither {$U-} nor {$C-} have been set, no buffer will be capable of being controlled. If these compiler options have been set, the **TypeAheadLegal** variable can be used to control keyboard buffering the way it is controlled in DOS programs.

The **TypeAheadLegal** variable does not, however, have any effect on keyboard buffering in the **ReadStr** procedure. The only way that buffering can be disabled in **ReadStr** is to use either the {$U+} or {$C+} compiler directives.

Initialization of global variables. In Turbo Pascal, an include-file module cannot initialize its own global variables, because there is no guarantee that any sin-

Non-stop flights to XENIX.

The XENIX® market is taking off. To keep up you've got to convert your MS-DOS® software to run on the new, faster 80286 machines. Quickly. Efficiently.

Nobody makes porting software to the XENIX environment as easy as Microsoft; you simply recompile the original source code.

Our full-featured XENIX languages use commands and syntax identical to their MS-DOS siblings. So you don't have to rewrite your program to move it.

Whether your applications are written in Microsoft versions of Pascal, BASIC, COBOL or FORTRAN, we have the solution to your transportation problems.

MICROSOFT®
The High Performance Software™

For the name of your nearest Microsoft dealer call (800) 426-9400. In Washington State, Alaska, Hawaii and Canada, call (206) 828-8088.

Overall features of XENIX Languages
- Shell support of syncronous and asyncronous tasks.
- Support of pipes for interprocess communication as a generalized I/O device.
- Record and file locking:
  - For sequential, random, dynamic and ISAM files.
  - Blocked or unblocked file reads.
  - Deny access, deny read and deny write lock levels.
  - Automatic or manual record and file locking of ISAM files.

Microsoft FORTRAN Compiler
- Based on ANSI 77 (x3.9-1978) standard.
- Single-and double-precision for both real and complex data types.
- Math packages: 287 support, alternative fast math and BCD decimal math.
- Large program support.

Microsoft COBOL Compiler
- High Level 2 ANSI 74 standard features.
- Multi-key ISAM with split keys, alternate keys and duplicate keys.
- Interactive extended screen section.

Microsoft BASIC Interpreter
- Standard Microsoft BASIC.
- Supports trace, single step and immediate execution.
- Supports random, dynamic, ISAM and sequential file I/O.

Microsoft Pascal Compiler
- ANSI and ISO standard with extensions.
- Math packages: 287 support, alternative fast math and BCD decimal math.
- Large program support.

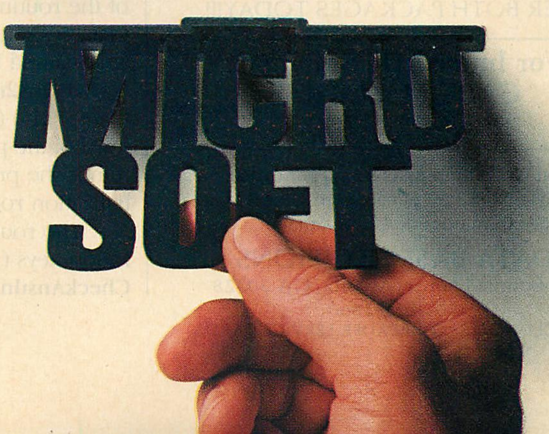


FIGURE 1: *BufLstBase*

```
VAR
  BufLstBase : BufferPtr;
  Dos2ioInitKey : REAL;

PROCEDURE InitDos2io;
BEGIN
  TypeAheadLegal := true;
  BufLstBase := nil;
  Dos2ioInitKey := 5721.0;
END; {InitDos2io}

PROCEDURE CheckInitialization;
BEGIN
  IF Dos2ioInitKey <> 5721.0 THEN
    abort('Please initialize with InitDos2io.');
END; {CheckInitialization}
```

BufLstBase must be initialized to zero, because it determines where an existing linked list of file buffers starts.

gle portion of its code will be called once and only once. This problem ranks just below the lack of separate compilation in the list of problems created by Turbo's lack of a true unit/module construct. **Dos2io** handles the difficulty in a way that may seem odd.

As an illustration of the problem, consider the global pointer **BufLstBase**, declared in **Dos2io-1.inc** (see figure 1). It is crucial to the operation of several parts of **Dos2io** that **BufLstBase** be initialized to **nil**, because **BufLstBase** is used to determine whether a linked list of file buffers has been created and where the list starts in the computer's memory. If **BufLstBase** has a value other than **nil**, the buffer allocation routine (**MakeBuffFor**) assumes that the list has been started at the address indicated by the value of **BufLstBase**, and it modifies several fields at that address and adds new buffers to the list. Imagine what this could do if **BufLstBase** started out pointing at a random address in memory—an address that could be within the operating system code itself.

Dos2io cannot initialize **BufLstBase** directly. The initialization key used in **Dos2io** and **AnsiStuf** forces the calling program to initialize the necessary variables by putting two routines—**InitDos2io** and **InitAnsiStuf**—in its main body. This technique works because all of the routines that reference global variables check the value of a real variable named **Dos2ioInitKey**.

If **Dos2ioInitKey** has a value other than 5721.0 (this value was chosen arbitrarily), the program aborts and reminds the programmer to call the initialization routines next time.

The routines that check the initialization keys (**CheckInitialization** and **CheckAnsiInit**) could not themselves call the initialization routines, because if a program were executed twice in a row, in the second execution the initialization key would falsely indicate that the initialization routine had been called. This happens because the compiler always allocates the same memory location to the initialization-key variable. The best solution is to have the program check the initialization key and abort (with a reminder) if it is not set.

Error handling. The boolean function **FlaggedErrors** determines whether or not the service routine has reported an error. The newer DOS functions set the carry flag if an error has occurred; because the carry flag is the least significant bit in the flags integer, **FlaggedErrors** must determine only whether the flags integer is odd.

To increase the readability of programs that use the **Dos2io** routines, most of those routines are actually functions that return the **ErrorMessage** data type. This data type gives programs a readable method of referring to the 18 standard DOS errors. The function **MessageType** converts the number used by the operating system to one of the following elements of the **ErrorMessage** type: **NoError**, **BadFunction**, **FileNotFound**, **PathNotFound**, **NoHandlesLeft**, **AccessDenied**, **BadHandle**, **MCBsDestroyed**, **TooLittleMemory**, **BadMemBlockAddr**, **BadEnvironment**, **BadFormat**, **BadAccessCode**, **BadData**, **MissingMessage**, **InvalidDrive**, **CurrentDirErase**, **DifferentDevice**, **NoMoreFiles**, **EndOfFile**, or **PartialRead**.

Dos2io was originally written as a Microsoft/IBM Pascal UNIT and preserves some of the Microsoft/IBM form to facilitate translation. The source code for **Dos2io-1.inc** and **Dos2io-2.inc** contains many more facilities than are described below. Programs can use all of the following routines, as well as any others defined at the top level of one of the include-file modules. The other procedures may be helpful for low-level work similar to the work that built **Dos2io**, but most users will not need them. Those who do should have no trouble understanding from the source code how the additional routines work.

To use the facilities of the entire library, a program must contain the following compiler directives somewhere near its beginning:

```
{$I dos2io-1.inc }
{$I dos2io-2.inc }
{$I dos2io-3.inc }
{$I AnsiStuf.inc }
```

Of course, if a programmer habitually includes all four files in every program,

PC WEEK'S PRODUCT OF THE YEAR
PC MAGAZINE'S AWARD FOR TECHNICAL EXCELLENCE

Borland Introduces the Laws of *TURBO DYNAMICS*™

Laws That Work Like Magic. Whether considering technological excellence, or innovation in areas such as pricing, not copy-protection, licensing agreements, site licenses, 60 day money-back guarantee —Borland is clearly recognized as the software industry leader. The following three laws of "*Turbo Dynamics*"™ exemplify our pledge for excellence.

1ST LAW:
SPEED, POWER AND PRICE.
Borland products are known to be fast, powerful and to deliver an incredible price performance ratio. We only believe in absolutely superb software at rock bottom prices.

2ND LAW:
NOT COPY-PROTECTED SOFTWARE AND REASONABLE LICENSING AGREEMENTS.
We will always offer not copy-protected versions of our software. Also, our licensing agreement is now so simple that even a child can understand it.

3RD LAW:
60 DAY MONEY-BACK GUARANTEE.
This third law is actually a first in the industry! We are so sure that you will love our software that all of our products now come backed with a 60 day money-back guarantee. No questions asked.

Turbo Dynamics Applies to Turbo Pascal. Borland's Pascal family of products is growing by leaps and bounds. You can now join hundreds of thousands of users and enter the world of Turbo Pascal programming. And remember, all three laws of *Turbo Dynamics* apply to all Borland products.

TURBO PASCAL™ $69.95

The industry standard. With more than 350,000 users worldwide Turbo Pascal is the industry's de facto standard. Turbo Pascal is praised by more engineers, hobbyists, students and professional programmers than any other development environment in the history of microcomputing. And yet, Turbo Pascal is simple and fun to use. **Free spreadsheet** included on every Turbo disk with ready-to-compile source code. ***Options:*** We offer the exciting Binary Coded Decimal (BCD) option for your business applications as well as an 8087 option for your number-crunching applications at a very low charge. Please refer to the coupon. ***Portability.*** Turbo Pascal is available today for most computers running PC-DOS, MS-DOS, CP/M-80 or CP/M-86. ***Jeff Duntemann, PC Magazine:*** "In its simplicity it achieves an elegance that no other language compiler has ever displayed."

TURBO GRAPHIX TOOLBOX™ $54.95

High resolution monochrome graphics and window management for the IBM PC. The Turbo Graphix Toolbox will give even a beginning programmer the expert's edge. It's a complete library of Pascal procedures and functions. Tools that will allow you to draw and hatch pie charts, bar charts, circles, rectangles and a full range of geometric shapes. Procedures that will save and restore graphic images to and from disk. And much, much, more. You may incorporate part or all of these tools in your programs and yet we won't charge you any royalties. Best of all, these functions and procedures come complete with commented source code on disk ready to compile.

TURBO TUTOR™ $34.95

From start to finish in 300 pages. Turbo Tutor is for everyone from novice to expert. Even if you've never programmed before Turbo Tutor will get you started right away. ***A must.*** You'll find the source code for all the examples in the book on the accompanying disk ready to compile. Turbo Tutor might be the only reference on Pascal and programming you'll ever need.

TURBO DATABASE TOOLBOX™ $54.95

The Turbo Database Toolbox is the perfect complement to Turbo Pascal. It contains a complete library of Pascal procedures that allows you to sort and search your data and build powerful applications. It's another Borland set of tools that will give the beginning programmer the expert's edge. ***Get started right away: free database!*** Included on every Toolbox disk is the source code to a working data base which demonstrates how powerful and easy to use our search system, Turbo-Access, really is. Modify it to suit your individual needs or just compile it and run. ***Remember, no royalties!***

NOT COPY-PROTECTED

TURBO PASCAL FAMILY

Available at better dealers nationwide. Call (800) 556-2283 for the dealer nearest you. To order by Credit Card call (800) 255-8008, CA (800) 742-1133

Carefully Describe your Computer System!

Mine is: ___ 8 bit ___ 16 bit
I Use: ___ PC-DOS ___ MS-DOS ___ CP/M 80 ___ CP/M 86

My computers' name/model is: ___

The disk size I use is: ☐ 3½" ☐ 5¼" ☐ 8"

Name: ___
Shipping Address: ___
City: ___
State: ___ Zip: ___
Telephone: ___

60 DAY MONEY-BACK GUARANTEE

Pascal 3.0	$ 69.95 ___
Pascal w/8087	$109.90 ___
Pascal w/BCD	$109.90 ___
Pascal w/8087 & BCD	$124.95 ___
Turbo Database Toolbox	$ 54.95 ___
Turbo Graphix	$ 54.95 ___
Turbo Tutor	$ 34.95 ___

*These prices include shipping to all U.S. cities. All foreign orders add $10 per product ordered.

Amount: (CA 6% tax) ___
Payment: VISA MC BankDraft Check
Credit Card Expir. Date: ___/___
Card #: ___

COD's and Purchase Orders WILL NOT be accepted by Borland. California residents: add 6% sales tax. Outside USA: add $10 and make payment by bank draft, payable in US dollars drawn on a US bank.

F6

BORLAND INTERNATIONAL

4585 Scotts Valley Drive, Scotts Valley CA 95066
Phone (408) 438-8400 Telex 172373

CIRCLE NO. 250 ON READER SERVICE CARD

FIGURE 2: *WriteStr*

```
PROCEDURE WriteStr(FileHandle : INTEGER; TheStr :
    dos2str255 );
VAR
  rgstr : regpack;
BEGIN {WriteStr}
  IF FileHandle = inp
   THEN abort('Cannot write to standard input.');
  with rgstr DO BEGIN
    b.x := FileHandle;
    c.x := length( TheStr );
    MakeAsciiZ( TheStr );
    ds := seg( TheStr );
    d.x := ofs( TheStr );
    a.h := $40;{Write to a file or device command}
    msdos( rgstr );
    IF FlaggedError( flags )
      THEN PrintMessage( MessageType( a.x ) );
    IF a.x < c.x  {if fewer than c.x bytes}
                  {were written}
      THEN abort( 'No room to write.' );
  END; {with rgstr}
END;  {WriteStr}
```

WriteStr is Dos2io's replacement for Turbo's intrinsic Write procedure.

his .COM files will be needlessly bloated. He could use three strategies to minimize the size of object code. First, he could delete unused routines from the modules when his program is in final form. Both modules avoid interprocedural dependencies when possible, so it is not difficult to determine which routines can be cut.

Second, he could overlay infrequently used module routines. This approach works best in extremely large applications; a program that calls a greater percentage of the library's routines cannot save as much space by cutting.

The third approach is the easiest—he could simply rely on the functional divisions between the include files.

DOS2IO-1.INC

Function IntStr (**TheNumber**: Integer; **StrLngth**: Integer): **dos2str80**;
IntStr returns a string representing **TheNumber**. It lets the programmer conveniently write integers with **WriteStr**. The convenience of **Write's** optional and multiple parameters can be approximated by using **concat** or Turbo's string-combining operator ("+"). The second parameter, **StrLngth**, determines the minimum size of the string that is returned.

Function RealStr (**TheNumber**: Real; **StrLngth**, **DigitsAfterDecimal**: Integer): **dos2str80**;
RealStr returns a string representing **TheNumber**. The names of the second and third parameters suggest the manner in which they control how the string is formatted.

Procedure WriteStr (**FileHandle**: Integer; **TheStr**: **dos2str255**);
WriteStr (see figure 2) is **Dos2io**'s replacement for Turbo's intrinsic **Write** procedure. Unlike **Write**, **WriteStr** can be used with files that are not in the current directory, allows command-line redirection of output, and is hardware-independent—that is, it functions the same on every DOS machine and does not need terminal installation. DOS predefines these five handles, which need not be opened before use:

0: Standard input device. Input can be redirected.
1: Standard output device. Output can be redirected.
2: Standard error output device. Output cannot be redirected.
3: Standard auxiliary device.
4: Standard printer device.

Dos2io defines the constant **outp** as 1 for the programmer's convenience in doing normal screen output. To use **WriteStr** for output to files other than those predefined by DOS, a **FileHandle** must be obtained from **OpenFile** or **CreateFile**, which are discussed below. The following are examples:

WriteStr(outp, 'Hello.');
WriteStr(FileHandle, StrVariable);

Procedure WriteEol (**FileHandle**: Integer);
WriteEol is **Dos2io**'s replacement for Turbo Pascal's intrinsic **WriteLln**. Programmers who use the **Write** procedures extensively may want to write a routine that combines **WriteStr** and **WriteEol**—something like Pascal's **WriteLn(string)**.

Procedure ReadStr (**FileHandle**: Integer; Var **TheStr**: **dos2str255**);
ReadStr (see figure 3) is **Dos2io**'s replacement for Turbo's intrinsic **ReadLn** procedure. It can be used with files that are not in the current directory, allows redirection of input, and is terminal-independent. With these exceptions, **ReadStr** operates exactly as does Turbo's **ReadLn**. **Dos2io** defines the integer constant **inp** for the programmer's convenience in doing normal, redirectable keyboard input. This example loads into the variable **YourString** all characters typed from the keyboard (unless input is redirected) until the user presses Return; the carriage-return/linefeed that is generated by Return is echoed but not stored in the string:

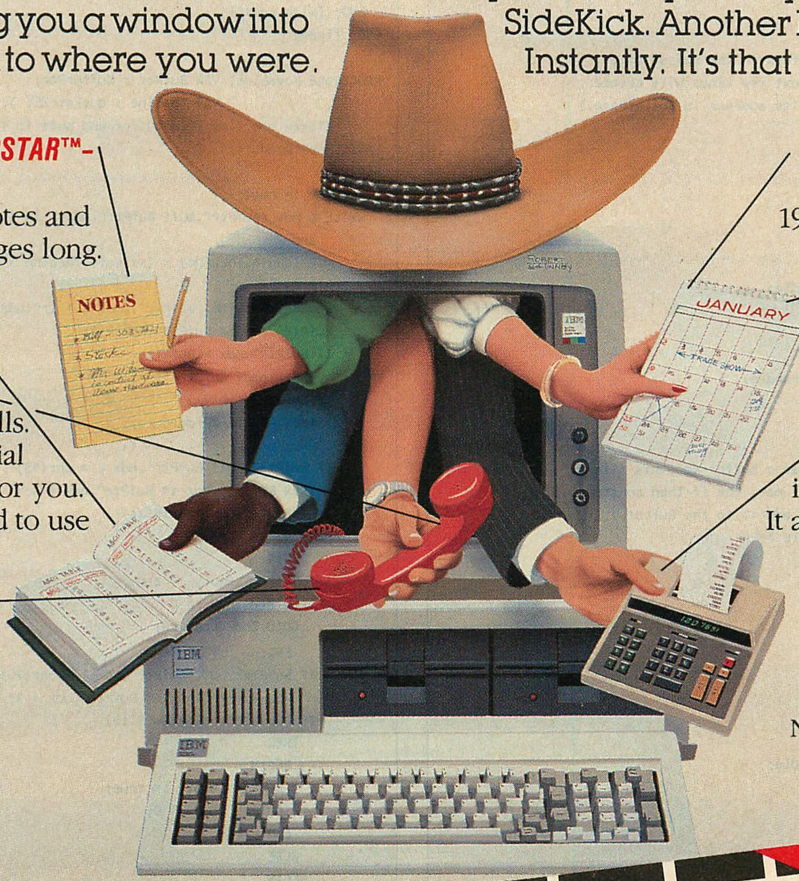
ReadStr(inp, YourString);

The following example, on the other hand, loads **YourString** with all characters in the file corresponding to **InputFileHandle** up to the next carriage return or EOF character. It does not store

INFOWORLD'S SOFTWARE PRODUCT OF THE YEAR

Borland's SideKick™ Will Clear Your Desk In 30 Minutes And Increase Your Productivity By 50%

SideKick is a combination of seven desktop accessories, which makes SideKick the single most effective business tool. Just a keystroke suspends your application program, giving you a window into SideKick. Another keystroke brings you back to where you were. Instantly. It's that easy.



A FULL-SCREEN WORDSTAR™-LIKE EDITOR
You may jot down notes and edit files up to 25 pages long.

AN ASCII TABLE
for easy reference.

AN AUTODIALER
for all your phone calls. It will look up and dial telephone numbers for you. (A modem is required to use this function.)

A PHONE DIRECTORY
for your names, addresses and telephone numbers. Finding a name or a number becomes a snap.

A MONTHLY CALENDAR
functional from year 1901 through year 2099.

A DATEBOOK
to remind you of important meetings and appointments.

A FULL-FEATURED CALCULATOR
ideal for business use. It also performs decimal to hexadecimal to binary conversions.

COPY-PROTECTED
$54.95

NOT COPY-PROTECTED
$84.95

THE CRITICS' CHOICE

"SIDEKICK deserves a place in every PC." **Garry Ray, PC WEEK**

"If you use a PC, get SIDEKICK. You'll soon become dependent on it." **Jerry Pournelle, BYTE**

"SIDEKICK is by far the best we've seen. It is also the least expensive." **Ron Mansfield, ENTREPRENEUR**

"In a simple, beautiful implementation of WordStar's™ block copy commands, SIDEKICK can transport all or any part of the display screen (even an area overlaid by the notepad display) to the notepad." **Charles Petzold, PC MAGAZINE**

BORLAND INTERNATIONAL
4585 Scotts Valley Drive, Scotts Valley CA 95066
Phone (408) 438-8400 Telex 172373
CIRCLE NO. 251 ON READER SERVICE CARD

NOT COPY-PROTECTED

SIDEKICK

Available at better dealers nationwide. Call (800) 556-2283 for the dealer nearest you. To order by Credit Card call (800) 255-8008, CA (800) 742-1133

Yes, I want the Best.
Please send me SideKick!

Copy-Protected **$54.95**
(CA res. add $3.30 tax per copy)
Quantity: ____ at $54.95

Not Copy-Protected **$84.95**
(CA res. add $5.10 tax per copy)
Quantity: ____ at $84.95

Amount: (CA 6% tax) ____

Payment: VISA MC BankDraft Check

Credit Card Expir. Date: ____/____

Card #: ____

S6

60 DAY MONEY-BACK GUARANTEE

Software for your IBM®PC, XT, AT, jr. and true compatibles.

PCjr requires not copy-protected version

These prices include shipping to all US cities. All foreign orders add $10 per product ordered.

Name: ____

Shipping Address: ____

City: ____

State: ____ Zip: ____

Telephone: ____

COD's and Purchase Orders WILL NOT be accepted by Borland. California residents: add 6% sales tax
Outside USA: add $10 and make payment by bank draft, payable in US dollars drawn on a US bank

FIGURE 3: *ReadStr*

```
PROCEDURE ReadStr( FileHandle : INTEGER;
                   VAR TheStr : dos2str255 );
LABEL EndProcedure;
VAR
  BufPtr : BufferPtr;

  PROCEDURE MakeBuffFor( FileHandle : INTEGER );
  LABEL
    EndProcedure;
  VAR
    OldPtr, TmpPtr : BufferPtr;
  BEGIN
    IF BufLstBase = nil THEN
      BEGIN
        {If there are no file buffers in the BufLst (i.e., if
        BufLstBase is nil), the next few lines will create
        the first buffer and put its address in BufLstBase.}
        new( TmpPtr );
        BufLstBase := TmpPtr;
        TmpPtr^.ndx := 1;
        TmpPtr^.eof := false;
        TmpPtr^.empty := true;
        TmpPtr^.next := nil;
        TmpPtr^.prev := nil;
        TmpPtr^.handle := FileHandle;
        GOTO EndProcedure;
      END;
    TmpPtr := BufLstBase;
    WHILE (TmpPtr^.handle <> FileHandle)
          and
          (TmpPtr^.next <> nil)
      DO TmpPtr := TmpPtr^.next;
        {This checks to see whether a buffer for this file is
        already in the BufLst.  If not, the if-then construct
        below will create one and add it to the BufLst.}
    IF TmpPtr^.handle <> FileHandle THEN
      BEGIN
        OldPtr := TmpPtr;
        new( TmpPtr );
        OldPtr^.next := TmpPtr;
        TmpPtr^.prev := OldPtr;
        TmpPtr^.ndx := 1;
        TmpPtr^.eof := false;
        TmpPtr^.empty := true;
        TmpPtr^.next := nil;
        TmpPtr^.handle := FileHandle;
      END;
    EndProcedure:
    BufPtr := TmpPtr;
  END;  {MakeBuffFor}

  PROCEDURE load( VAR BufPtr : BufferPtr );
  VAR
    rgstr : regpack;
  BEGIN
    fillchar( BufPtr^.buf, sizeof( BufPtr^.buf ), chr(0) );
    with rgstr DO BEGIN
      b.x := FileHandle;
      c.x := BufSize;
        {CX gets the number of bytes to be transferred.}
      ds := seg( BufPtr^.buf );
      d.x := ofs( BufPtr^.buf );
      a.h := $3F;
        {Read from a file or device command.}
      msdos( rgstr );
      IF FlaggedError( flags )
        THEN PrintMessage( MessageType( a.x ) );
      BufPtr^.empty := a.x <= 2;
        {AX contains the number of bytes actually transferred.
        If the value is zero, the program has tried to read
        from the end of file.  If the value is less than or
        equal to 2, we've probably read a carriage return
        from the keyboard.}
      IF BufPtr^.empty
        THEN
          BEGIN
            BufPtr^.ndx := 0;
            BufPtr^.eof := a.x = 0;
          {Should never get here with normal files, because
          MoveLine will set eof when it finds chr(26). However,
          not all files use chr(26) as an eof marker.}
          END
        ELSE
          BEGIN
            BufPtr^.ndx := 1;
            BufPtr^.eof := false;
          END;
    END; {with rgstr}
  END; {load}

  PROCEDURE MoveLine( VAR BufPtr : BufferPtr;
                      VAR TheLine : dos2str255 );
      {Takes one line from buffer and puts it in TheLine.}
  LABEL EndProcedure;
  BEGIN
    TheLine := null;
    WHILE ( not (BufPtr^.buf[ BufPtr^.ndx ] in [ #13, #26, #0 ]))
          and
          ( length(TheLine) < (sizeof( TheLine ) - 2)) DO
      BEGIN
        AddStr( TheLine, BufPtr^.buf[ BufPtr^.ndx ] );
        BufPtr^.ndx := BufPtr^.ndx + 1;
        IF BufPtr^.ndx > BufSize
          THEN load( BufPtr );
        IF BufPtr^.empty
          THEN GOTO EndProcedure;
        END;
        IF BufPtr^.buf[ BufPtr^.ndx ] = chr(13)
          THEN BufPtr^.ndx := BufPtr^.ndx + 1;
        IF (BufPtr^.ndx > BufSize)
          THEN
           BEGIN
            BufPtr^.empty := true;
            BufPtr^.ndx := 0;
            GOTO EndProcedure;
            END;
          IF BufPtr^.buf[ BufPtr^.ndx ] = chr(10)
            THEN BufPtr^.ndx := BufPtr^.ndx + 1;
          IF (BufPtr^.ndx > BufSize)
            THEN
             BEGIN
              BufPtr^.empty := true;
              BufPtr^.ndx := 0;
              END
            ELSE
           BEGIN
            IF (BufPtr^.buf[ BufPtr^.ndx ] = chr(0))
             THEN
              BEGIN
               BufPtr^.empty := true;
               BufPtr^.ndx := 0;
              END
             ELSE
              IF BufPtr^.buf[ BufPtr^.ndx ] = chr(26)
               THEN BufPtr^.eof := true;
          END;
    EndProcedure:
  END;  {MoveLine}

BEGIN {ReadStr}
  CheckInitialization;
  IF FileHandle = outp
    THEN abort( 'Cannot read from standard output.' );
  TheStr := null;
  MakeBuffFor( FileHandle );
  IF BufPtr^.empty
    THEN load( BufPtr );
  IF not BufPtr^.empty
    THEN MoveLine( BufPtr, TheStr );
END;  {ReadStr}
```

Procedure ReadStr can be used with files that are not in the current directory; it allows redirection of input.

FIGURE 4: *GetKey*

```
PROCEDURE GetKey( VAR ch:CHAR; ReturnOnMatch:
    dos2charset );
VAR
  rgstr : regpack;
BEGIN {GetKey}
  CheckInitialization;
  IF TypeAheadLegal
        {Works only if U- and C- compiler
      directives are set in all files.}
    THEN rgstr.A.H := 8
        {Console input without echo.}
    ELSE rgstr.A.H := $C;
        {Clear standard input buffer and
      invoke input function stored in AL.
        See D-20 of PC-DOS 2.0 manual.}
  REPEAT
    rgstr.A.L := 8;
    msdos( rgstr );
    ch := CHR(rgstr.A.L);
  UNTIL ch IN ReturnOnMatch;
END; {GetKey}
```

GetKey waits for one of the keys listed in ReturnOnMatch and stores it in ch without echoing it to the screen.

carriage-return/linefeed pairs in the string, does not store the EOF character, and echoes nothing to the screen:

ReadStr(InputFileHandle, YourString);

ReadStr does its own buffering when it reads from files. It creates a new buffer for each file when the buffer is needed and preserves it until the file is closed. If more than one file is being read, **ReadStr** will automatically allocate a unique buffer for each one, thus allowing efficient simultaneous input from multiple files.

Function EndReached (**FileHandle**: Integer): Boolean;

EndReached is **Dos2io**'s replacement for Turbo's intrinsic EOF function. It returns a True if **ReadStr** says it has reached the end of **FileHandle**. **EndReached** works only with **ReadStr**, because the operating system may conclude that the end of a file has been reached before **ReadStr** gets to the end of the file's buffer. A different function, **EndFile**, serves as an end-of-file function for nonbuffered file I/O.

Procedure GetKey (Var **ch**: Char; **ReturnOnMatch**: **dos2charset**);

GetKey's less powerful counterpart among the Turbo Pascal intrinsic procedures is the **Read** (KBD) command. **GetKey** (see figure 4) waits for one of the keys listed in **ReturnOnMatch** and stores it in **ch** without echoing it to the screen. **Dos2io** declares a character variable **Choice** primarily for the convenience of any routines that use either **GetKey** or **GetExtendedKey** (described below). This variable can be used in **GetKey**'s first parameter, saving the programmer from having to declare a special-purpose character variable in every routine that calls **GetKey** or **GetExtendedKey**. The use of set notation in the second parameter in order to list the options approximates the convenience of having a variable number of parameters. In the following example, **GetKey** waits for the user to press Return (#13), Esc (#27), or any other key that produces a printable character in the ASCII character set:

GetKey(choice, [**#13, #27,** ' ' .. '~']);

These routines use DOS service routine 8H (or OCH if **TypeAheadLegal** is False); they have been made compatible with DOS 2.0's I/O redirection system.

Procedure GetExtendedKey (Var **ch**: Char; **legalchars**: **dos2charset**; **legalxkeys**: **dos2numset**; Var **xkeygotten**: Boolean);

GetExtendedKey has no counterpart in Turbo's intrinsic procedures. It resembles **GetKey**, but it allows special keys, such as arrow and function keys and Ctrl-key or Alt-key combinations, to be among the user's choices. Extended code values for such special keys are listed in the IBM BASIC manual and will differ on machines that do not claim IBM compatibility. The following example would wait for the user to press the spacebar, the Return key, or any one of the ten function keys:

```
GetExtendedKey ( choice,
[' ', chr(13)], [59..68],
SpecialKey
);
If SpecialKey Then
Case ord (choice) of
#72 : etc. ...
```

DOS2IO-2. INC

Procedure GetProgramParameter (Var **ParamStr**: **dos2str80**);

GetProgramParameter returns everything that was typed on the DOS command line upon invocation of the program except redirection symbols and the name of the program's .COM file. The operating system filters out redirection symbols to keep I/O redirection transparent to applications. Version 2.0 of Turbo Pascal discourages the development of applications driven from the DOS command line because it puts an automatic screen clear at the beginning of every .COM file the compiler generates and automatically sends the cursor to the bottom left corner of the screen (the EOS) at the end of each program. There is no good reason for this. DEBUG can be used to suppress both the

beginning screen clear and the ending cursor-to-EOS.

Function OpenFile (Var **FileHandle**: Integer; **fname**: **dos2str80**): **ErrorMessage**;

OpenFile (see figure 5) returns a DOS **FileHandle** for the file identified by **fname**. **FileHandle** can subsequently be used with other I/O routines in this library. **OpenFile** replaces some, but not all, of the functions served by Pascal's **Reset**. If **Reset** is performed on an already open file, the file pointer returns to the first element of the file. Because file handles have some unique features that deal with attempts by more than one program or user to access a file at the same time, **OpenFile** cannot work that way. To return the file pointer to the first element of a file identified by a handle, the **SetPtrFromStart** function (described below) must be used. Special notice should be taken that both **OpenFile** and **CreateFile** follow the UCSD Pascal convention of associating an internal file variable with an external file name and opening the variable for I/O in a single step. Although Turbo generally follows UCSD conventions, on this point it follows Microsoft Pascal instead and requires two steps. UCSD's approach works better here because of the division of duties between **OpenFile** and **SetPtrFromStart**. **Dos2io**, therefore, has nothing that corresponds to Turbo's **Assign** procedure.

Function CreateFile (Var **FileHandle**: Integer; **fname**: **dos2str80**): **ErrorMessage**;

The **CreateFile** function (see figure 6) creates a file and returns a DOS **FileHandle** for that file that can be used with other I/O routines. It replaces Pascal's intrinsic **Rewrite** procedure.

Function CloseHandle (**FileHandle**: Integer): **ErrorMessage**;

CloseHandle serves roughly the same function as Pascal's intrinsic **Close** procedure, with the exception that the corresponding file handle need not be expressly closed for the name of a newly created file to remain in the disk directory. **CloseHandle** does, however, insure that the directory properly records the size, date, and time of the file. It also frees up memory that the operating system allocates to the file handle. **FileHandles** should be closed when they are no longer needed. The number of **FileHandles** that can be used is set in the CONFIG.SYS file at boot time with the FILES command. The maximum is 99 and the default is 8, so it is easy to run out of handles if many files are open simultaneously. Most versions of Pascal, including Turbo, automatically close a file upon exiting the block in which the file was declared. File handles must be closed explicitly; they are not closed automatically, even when the program terminates.

FIGURE 5: *OpenFile*

```
FUNCTION  OpenFile ( VAR FileHandle : INTEGER;
                     fname : dos2str80): ErrorMessage;
VAR
  rgstr : RegPack;
BEGIN  {OpenFile}
  FileHandle := 0;
  OpenFile := NoError;
  MakeAsciiZ( fname );
  WITH rgstr DO BEGIN
    a.h := $3D;     {open a file}
    A.L := 2;       {for reading and writing}
    b.x := 0;
    c.x := 0;
    d.x := ofs( fname );
    ds := seg( fname );
    msdos( rgstr );
    IF FlaggedError( flags )
      THEN OpenFile := MessageType( a.x )
      ELSE FileHandle := a.x;
  END; {WITH rgstr}
END;  {OpenFile}
```

The FileHandle returned by OpenFile is used with other routines in Dos2io.

FIGURE 6: *CreateFile*

```
FUNCTION  CreateFile ( VAR FileHandle : INTEGER;
                     fname : dos2str80): ErrorMessage;
VAR
  rgstr : RegPack;
BEGIN  {CreateFile}
  FileHandle := 0;
  CreateFile := NoError;
  MakeAsciiZ( fname );

  WITH rgstr DO BEGIN
    a.h := $3C;     {create a file}
    A.L := 0;
    b.x := 0;
    c.x := 0;   {attribute of the file;
                 0 makes it normal}
    d.x := ofs( fname );
    ds := seg( fname );
    msdos( rgstr );
    IF FlaggedError( flags )
      THEN CreateFile := MessageType( a.x )
      ELSE FileHandle := a.x;
  END; {WITH rgstr}
END;  {CreateFile}
```

CreateFile creates a file and returns a DOS FileHandle for that file; CreateFile replaces Turbo's Rewrite procedure.

Function SetFileMode (**FileName**: **dos2str80**; **TheSet**: **ModeSet**): **ErrorMessage**;

The user can pass **FileName** a string with a drive, path, and file name and then use set notation to pass **TheSet** one or more of the four elements of the data type **FileMode**. **SetFileMode** changes the attribute of the file associated with **FileName**. The elements of type **FileMode** are: **NormalFile**, **ReadOnlyFile**, **HiddenFile**, and **SystemFile**. **NormalFile** cancels the other three, but any combination of **ReadOnlyFile**, **HiddenFile**, and **SystemFile** may be used. Some examples of the use of function **SetFileMode** are:

printmessage(SetFileMode
  ('b:letter2.txt', [ReadOnlyFile,
  HiddenFile]));
printmessage(SetFileMode
  ('a:IBMBIO.COM', [NormalFile]));

Function GetFileMode (**FileName**: **dos2str80**; Var **TheSet**: **ModeSet**): **ErrorMessage**;

Pass **FileName** a string with a drive, path, and file name, and pass **TheSet** a variable of type **ModeSet**. If **GetFileMode** finds the file that corresponds with **FileName**, it will store all of the file's attributes in **TheSet**; if it does not find the file or finds it but is denied access to it, **GetFileMode** returns an **ErrorMessage** and sets **TheSet** to [ ], the empty set. For example:

PrintMessage (GetFileMode
  ('a:IBMBIO.COM', YourModeSet));
If YourModeSet =
  [ReadOnlyFile, HiddenFile, SystemFile]
Then
  WriteStr (outp, 'Just as expected.')
Else
  WriteStr (outp, 'Something amiss.');

Function FileLength (**FileHandle**: Integer): Real;

FileLength returns the number of bytes in the file identified by **FileHandle**.

Function RenameFile (**OldFileName**, **NewFileName**: **Dos2str80**): **ErrorMessage**;

A string that contains the drive, path, and file name of the file to be renamed should be passed into **OldFileName**. A string that contains the path and file name to which the file is to be renamed should be passed into **NewFileName**. If a drive is used in the second string, it must be the same as the drive that was specified or implied in the first string. This is because function **RenameFile** does not permit the transfer of files between disks. However, the directory paths do not have to be the same, which allows a file to be moved to another directory and renamed (if necessary) in the process.

Function DeleteFile (**FileName**: **Dos2str80**): **ErrorMessage**;

Pass **FileName** a string containing a drive, path, and file name. Global file name characters are not allowed in any part of the string. **DeleteFile** removes the directory entry associated with **FileName**. It will not delete read-only

More Power Than You Thought Possible

Arity offers the first serious implementation of Prolog for IBM personal computers. Arity/Prolog is a powerful, highly optimized, and extended version of the logic programming language Prolog. Imagine building software applications with a language that solves problems through deduction and logical inference. The task of creating complex programs is much faster and easier, resulting in lower development costs. Arity/Prolog is now in use in a wide range of applications in industry, business, research, and education. The solution— the *Arity/Prolog Interpreter:*

- Source level debugger
- Virtual databases, each with a workspace of 16 megabytes
- Floating-point arithmetic
- String support for efficient text handling

- Interface to assembly language and 'C'
- Text screen manipulation
- Integrated programming shell to MSDOS
- Comprehensive set of evaluable predicates
- Definite clause grammar support

Arity/Prolog Interpreter $495.00

Arity also offers the *Arity/Prolog Compiler and Interpreter,* a sophisticated development environment for building AI applications. Essential for producing fast, serious production code.

Arity/Prolog Compiler and Interpreter $1950.00

The *Arity/Prolog Demo Disk* is available for $19.95. ■ Arity/Prolog products run on the IBM PC, XT, AT, and all IBM compatibles. ■ To order, call (617) 371-2422 or use the order form below.

arity corporation 358 Baker Avenue, Concord, MA 01742

Name

Organization

Address

☐ Enclosed is a check or money order to Arity Corporation

☐ Please bill my ☐ Mastercard ☐ Visa ☐ American Express

Account #

Valid ___/___ to ___/___

signature

Quantity	Product	Unit Price	Total Price
	Arity/Prolog Compiler & Interpreter	$1950.00	
	Arity/Prolog Interpreter	$ 495.00	
	Arity/Prolog Demo Disk	$ 19.95	
		Subtotal	
		MA residents add 5% sales tax	
		Total Amount	

☐ Please send me more information about Arity and Arity/Prolog

arity 358 Baker Avenue, Concord, MA 01742

M-AD-04

files. To delete a read-only file, first change its attribute using **SetFileMode**.

Function DefaultDrive: Char;

This function returns the current default drive as a lowercase letter.

Procedure SetDefaultDrive
(DriveLetter: Char);

SetDefaultDrive sets the default drive to the drive letter indicated by its parameter. The drive letter can be either upper- or lowercase.

Function BlockRead (FileHandle:
Integer; Var buffer: **buftype**;
BlockNumber: Integer):
ErrorMessage;

This version of **BlockRead** resembles UCSD Pascal's **BlockRead** function more than it does Turbo Pascal's. Turbo's **BlockRead** does not allow specification in its parameter list of which block is to be read. **Dos2io**'s version, like UCSD's, does not need sequential **BlockReads**. Block 8 can be read before block 0, and then block 15 can be read, and so on; the third item in the parameter list is the block number to be read. Also, unlike Turbo's **BlockRead**, this version of the function works with file handles instead of file control blocks and therefore can be used with files not in the current directory. The size of the block used in **BlockRead** and **BlockWrite** (described below) is determined by the integer constant **Buf Size**. **Buf Size** also controls the size of each file buffer that is maintained by **ReadStr**.

Function BlockWrite (FileHandle:
Integer; Var buffer: **buftype**;
BlockNumber: Integer):
ErrorMessage;

BlockWrite is essentially the converse of **BlockRead**, but with an important difference. **BlockWrite** cannot guarantee that data will be written within the current file boundaries.

Procedure SetPtrFromStart (FileHandle:
Integer; **OffSetFromStart**: Real);

SetPtrFromStart allows the programmer to select the point at which a read or write operation on a file is to begin. Its counterparts among the Turbo intrinsic procedures are **Seek** and **Longseek**. Use this procedure to reset an open file. Pass 0 to **OffSetFromStart** (the second parameter) and the file pointer will move back to the first record (that is, the first byte) of the file. Remember that if **ReadStr** is maintaining an input buffer for the file handle passed into either **SetPtrFromStart** or **SetPtrFromEnd** (described below), these procedures will automatically flush **ReadStr**'s buffer. This insures an accurate relationship between the operating system's file pointer and **ReadStr**'s pointer into its buffer for the file.

Procedure SetPtrFromEnd (FileHandle: Integer; OffSetFromEnd: Real);

SetPtrFromEnd is essentially the same as **SetPtrFromStart**, except that the file pointer can be set relative to the end of the file. The value passed to **OffSetFromEnd** may be either negative or positive. A negative value positions the file's read/write pointer to some point *before* the end of the file, allowing bytes at the end of a file to be overwritten. If **OffSetFromEnd** is greater than 0, then the file will contain whatever happens to be on the disk in the sectors that the operating system allocates to the file as it grows.

Function EndFile (FileHandle: Integer): Boolean;

EndFile is an end-of-file function for use with nonbuffered file I/O. Notice that it does not work with **ReadStr**; the **ReadStr** procedure does its own buffering, so the operating system's file pointer may be as many as 512 bytes ahead of **ReadStr**'s pointer into its buffer. **EndReached** can be used as an equivalent of the end-of-file for **ReadStr**.

Function MkDir (DirName: dos2str80): ErrorMessage;

Pass **MkDir** a string with drive and directory path names. If any member of the directory path does not exist, the directory path is not changed. Otherwise, a new directory is created at the end of the specified path.

Function RmDir (DirName: dos2str80): ErrorMessage;

Pass **RmDir** a string with drive and directory path names. The specified directory is removed from the structure. The current directory cannot be removed. Note that the **AccessDenied** error message is returned if the specified directory is not empty.

Function ChDir (DirName: dos2str80): ErrorMessage;

Pass **ChDir** a string with drive and directory path names. If any member of the directory path does not exist, the directory path is not changed. Otherwise, the current directory is set to the string.

Procedure SetDate (month, day, year: Integer);

Procedure GetDate (Var month, day, year: Integer);

These two procedures set and return the system date in the manner suggested by the variable names in their parameter lists. **SetDate** ensures that the month and day are in the ranges 1 to 12 and 1 to 31, respectively. **GetDate** returns month and day values in these ranges. **SetDate** will accept year values in the ranges 80 to 199 or 1980 to 2099. **GetDate** returns year values in the range 1980 to 2099.

HOW ANSISTUF WORKS

Turbo Pascal's intrinsic screen-handling capabilities can be greatly extended by using a program called **AnsiStuf**. More importantly, **AnsiStuf** allows a programmer to develop an application that will run without recompilation or terminal installation on any machine using MS-DOS 2.0 or later on which the ANSI.SYS driver (which is supplied with DOS 2.0 or later) can be installed, regardless of the IBM compatibility of that machine. **AnsiStuf** does not force a program to settle for the slow performance of the ANSI.SYS device on a PC-compatible computer. Depending on the IBM compatibility of the specific machine, a user can set a simple operating-system parameter that will make the program run as fast as (or even faster than) Turbo's intrinsic screen-handling routines.

AnsiStuf provides a solution to the dilemma facing programmers who need high-performance screen-handling but who also want to write for the entire MS-DOS market. A program that uses **AnsiStuf** will run identically on *any* MS-DOS machine, although its speed

will vary greatly among machines.

A program that handles screen output through **AnsiStuf** saves its users the inconvenience of terminal installation and prevents the programmer from having to choose among screen-handling methods that strike different balances between speed and transportability. **AnsiStuf** is not simply an ANSI.SYS screen driver. The programmer can develop one program that will adopt the speediest method of screen handling for the machine on which it is run.

On 100-percent IBM-compatible machines, an **AnsiStuf**-based program writes directly into the display adapter's memory. On machines such as the Tandy 2000 and the TI Professional, which are IBM-compatible only at the ROM BIOS level, a program based on **AnsiStuf** adopts Turbo's screen-handling method—direct calls to the interrupt 10H service routines. On machines that do not purport to be IBM-compatible at all, such a program uses ANSI.SYS and string-oriented output through the operating system. Full-screen control is implemented identically in all three modes; the programmer need pay little or no attention to the question of which screen-handling method the program will use at any particular time.

The screen-control method to be used is not set at compile time and may be changed at will without altering a single byte of the program's code. A program's user controls its screen-handling method by employing the DOS SET command to change the parameter of an environment string named **VideoMethod**. If no **VideoMethod** parameter is set, or if the parameter is set to ANSI, the program will use ANSI.SYS and string-oriented output through the operating system. Full control over text colors and/or monochrome attributes is thereby preserved. If, however, the user is working on a machine that is only moderately IBM-compatible, and he types in the command

```
A>set videomethod=rom
```

before executing the program, the program will use direct calls to the IBM ROM BIOS for all screen output. It will do its screen-handling much more rapidly and its output will not be redirectable, but in all other respects the program will operate exactly as it does when the **VideoMethod** environment string is set to ANSI.

A user working on an extremely IBM-compatible machine can type

```
A>set videomethod=dma
```

on the DOS command line and boost the speed of the program's screen-handling facility by any order of magnitude. In its DMA mode, **AnsiStuf** writes directly into the display adapter's memory. With the exception of its greatly increased speed, the program will still operate exactly as it does when **VideoMethod** is set to ROM or ANSI.

AnsiStuf allows the programmer to be largely unconcerned with how the screen display will be handled. It requires only the following adaptations:

- The program must confine its I/O to facilities provided in **AnsiStuf** (or in **Dos2io**, where using ROM or DMA screen-handling is not important).
- Because DMA screen-handling does not move the cursor, **AnsiStuf** requires an additional step to guarantee consistent cursor placement between video methods. In all **AnsiStuf** routines in which the cursor might move in ANSI or ROM modes but not in DMA mode, a pair of global integer variables named **NominalCol** and **NominalRow** must be updated. A program should execute **SC_GotoXY (NominalCol, NominalRow)** after

mbp COBOL for your IBM/PC

Now, more convenient... 4 (exclusive) ways.

Now, the mbp COBOL Compiler offers unrivaled convenience to go with its unmatched performance.

Here are the convenience features you've wished for:

1) an enhanced Screen Management System with program-controlled video attributes and color; **2)** support for PATH & sub-directories; **3)** DOS command execution from within a COBOL program; **4)** 'permanent' DEFAULT modification.

The new mbp Compiler has them all! And they're exclusives: you get them *only* with mbp.

GIBSON MIX Benchmark Results Calculated S-Profile *(Representative COBOL statement mix)* Execution time ratio			
mbp* COBOL	Level II* COBOL	R-M* COBOL	Microsoft* COBOL
1.00	4.08	5.98	6.18

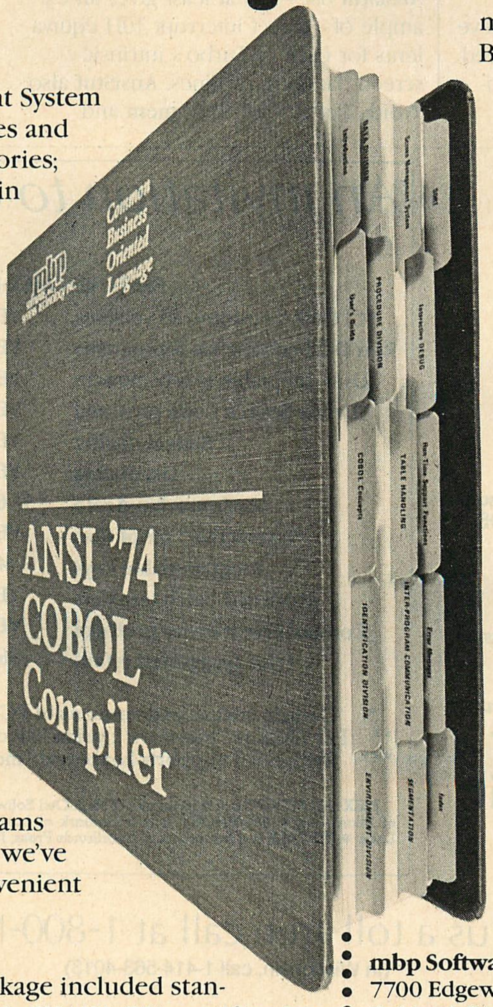
*128K system with hard disk required. IBM PC & AT are IBM TMs; Netware is a Novell, Inc. TM; Level II is a Micro Focus TM; R-M is a Ryan-McFarland TM; Microsoft is a Microsoft TM.

Plus, it's 4 times faster.

Because the mbp COBOL Compiler generates native machine language object code, it executes programs *at least* 4 times faster (see chart). Now, we've made that performance even more convenient to use.

The complete COBOL.

An Interactive Symbolic Debug Package included standard; Multi-keyed ISAM structure; SORT & CHAIN; GSA certification to ANSI '74 Level II; IBM/PC-AT and TI Professional compatibility; with mbp, you get it all. Optional: Novell NetWare interface.



mbp COBOL: the choice of professionals.

It's no surprise more and more companies like Bechtel, Bank of America, Chase, Citicorp, Connecticut Mutual, Hughes Aircraft, McDonnell-Douglass, and Price-Waterhouse choose mbp COBOL.

Make it your choice, too. Just send the coupon, or call, for complete information. Today.

mbp COBOL. $1000

Please send complete mbp COBOL information to:

NAME ____________

COMPANY ____________

ADDRESS ____________

CITY/STATE/ZIP ____________

____

PHONE ____________

mbp Software & Systems Technology, Inc.
7700 Edgewater Drive, Suite 360
Oakland, CA 94621
Phone 415/632-1555

mbp®

CIRCLE NO. 172 ON READER SERVICE CARD

each unbroken sequence of **WriteAt** commands if cursor placement makes a difference. If cursor placement is unimportant, a program can turn the cursor off (using **CursorHeight**) for cosmetic purposes.

The programmer is responsible for guaranteeing that the user cannot select **VideoMethod DMA** in a graphics mode. DMA video-handling works only in text modes—in graphics modes it will spew strange patterns across the screen. In a program that uses graphics, the programmer may want to build in a way of disabling the DMA mode while the machine is in a graphics mode.

AnsiStuf does a great deal of error-checking to protect programmers unaccustomed to its use. The error-checking facilities help with debugging but serve no purpose in a finished, fully debugged application. They should be deleted at that point. Although they do not significantly reduce the speed of screen-handling in ANSI mode, the effect is noticeable in ROM mode and dramatic in DMA mode. Calls to the integer function **Between** should be deleted to achieve the easiest and most immediate speed gains. Applications that can independently guarantee consistent cursor placement between video methods might also achieve speed gains by removing all references to the variables **NominalRow** and **NominalCol**.

To execute a program that uses **AnsiStuf**'s default mode, the user must have a boot disk that includes a copy of Microsoft's ANSI.SYS; in addition, the CONFIG.SYS file on the boot disk must include the following line:

```
device=ansi.sys
```

Any use of the intrinsic I/O procedures in the PC-DOS version of the compiler will crash a machine for which the program has not been installed. The intrinsic I/O routines in the MS-DOS generic version of Turbo Pascal are unlikely to crash a machine but will not work without installation.

In anticipation of the need to move **AnsiStuf** into another language (specifically, into Modula-2 or back into Microsoft/IBM Pascal), **AnsiStuf** avoids using Turbo Pascal's nonstandard intrinsic procedures. Because a general interrupt gate, such as Turbo's **INTR** routine, should be easy to write in any language, **AnsiStuf** uses—or at least gives an example of—direct interrupt 10H equivalents for each of Turbo's intrinsic screen-handling routines. **AnsiStuf** also avoids Turbo's intrinsic **mem** and **MemW** arrays because their syntax is nonstandard. Instead, **AnsiStuf** uses the simple routines **PeekByte**, **PokeByte**, **PeekWord**, and **PokeWord**, which use a highly transportable method of directly accessing memory locations. These routines should work with little or no modification in Microsoft/IBM Pascal and in any implementation of Modula-2.

The routines available in the **AnsiStuf** library are discussed below.

Procedure sc_left;
Procedure sc_right;
Procedure sc_up;
Procedure sc_down;

These four procedures have no counterpart in Turbo's screen-control routines. They move the cursor one line or column in the direction indicated.

Procedure sc_GotoXY (**column, row**: Integer);

This is **AnsiStuf**'s replacement for Turbo Pascal's intrinsic **gotoxy** procedure. The first and second parameters control horizontal and vertical cursor placement. Like other **AnsiStuf** routines, this one numbers columns from the left, beginning at 1, and rows from the top, beginning at 1—as do the intrinsic screen-handling routines.

Procedure Read WithoutEcho (Var **thestr**: sc_str255);

ReadWithoutEcho gets a string from

Tools That Make Your Job Easier

For PCDOS/MSDOS (2.0 and above/128K) • IBM PC/Compatibles, PC Jr., Tandy 1000/1200/2000, & others
For CPM80 2.2/3.0 (Z80 required/64K) • 8″ SSSD, Kaypro 2/4, Osborne I SD/DD, Apple II, & others

MIX EDITOR

Programmable, Full/Split Screen Text Processor

Introductory Offer **29.95**

Great For All Languages

A general purpose text processor, the MIX Editor is packed with features that make it useful with any language. It has auto indent for structured languages like Pascal or C. It has automatic line numbering for BASIC (255 character lines). It even has fill and justify for English.

Split Screen

You can split the screen horizontally or vertically and edit two files simultaneously.

Custom Key Layouts

Commands are mapped to keys just like WordStar. If you don't like the WordStar layout, it's easy to change it. Any key can be mapped to any command. You can also define a key to generate a string of characters, great for entering keywords.

Macro Commands

The MIX Editor allows a sequence of commands to be executed with a single keystroke. You can define a complete editing operation and perform it at the touch of a key.

Custom Setup Files

Custom keyboard layouts and macro commands can be saved in setup files. You can create a different setup file for each language you use.

MSDOS Features

Execute any DOS command or run another program from inside the editor. You can even enter DOS and then return to the editor by typing exit.

MIX C COMPILER

Full K&R Standard C Language
Unix Compatible Function Library

Introductory Offer **39.95**

Complete & Standard

MIX C is a complete and standard implementation of C as defined by Kernighan and Ritchie. Coupled with a Unix compatible function library, it greatly enhances your ability to write portable programs.

The Best C Manual

MIX C is complemented by a 400 page manual that includes a tutorial. It explains all the various features of the C language. You may find it more helpful than many of the books written about C.

Fast Development

MIX C includes a fast single pass compiler and an equally fast linker. Both are executed with a simple one line command. Together they make program development a quick and easy process.

Fast Execution

The programs developed with MIX C are fast. For example, the often quoted prime number benchmark executes in a very respectable 17 seconds on a standard IBM PC.

Standard Functions

In addition to the functions described by K&R, MIX C includes the more exotic functions like *setjmp* and *longjmp*. Source code is also included.

Special Functions

MIX C provides access to your machine's specific features through BDOS and BIOS functions. The CHAIN function lets you chain from one program to another. The MSDOS version even has one function that executes any DOS command string while another executes programs and returns.

Language Features

- Data Types: char, short, int, unsigned, long, float, double (MSDOS version performs BCD arithmetic on float and double-no roundoff errors)
- Data Classes: auto, static, extern, register
- Struct, Union, Bit Fields (struct assignment supported)
- Typedef, Initialization
- All operators and macro commands are supported

30 DAY MONEY BACK GUARANTEE

Orders Only: **Call Toll Free 1-800-523-9520,** (Texas only **1-800-622-4070**)

MIX Editor ____ $29.95 + shipping ($5 USA/$10 Foreign) Texas residents add 6% sales tax
MIX C ____ $39.95 + shipping ($5 USA/$25 Foreign) Texas residents add 6% sales tax

Visa ____ MasterCard ____ Card # ________ Exp. Date ____
COD ____ Check ____ Money Order ____ Disk Format ________
Computer ________ Operating System: MSDOS ____ PCDOS ____ CPM80 ____
Name ________
Street ________
City/State/Zip ________
Country ________
Phone ________

MIX software
2116 E. Arapaho
Suite 363
Richardson, Tx 75081

Dealer Inquiries Welcome
Call (214) 783-6001

T

the keyboard without echoing it to the screen. This procedure probably will be used rarely in a direct fashion, owing to the fact that the operating system handles echo suppression automatically when it reads something other than the keyboard. However, the procedure has an important role in **sc_CursorPosition**.

Procedure sc_CursorPosition (Var row, column: Integer);

This procedure has two counterparts in Turbo's intrinsic routines: the **WhereX** and **WhereY** functions. It combines the functions into a single procedure because both ANSI.SYS and ROM BIOS service routine 3 return row and column values with a single operation. Separate functions for **WhereX** and **WhereY** would be redundant and slow.

ANSI.SYS returns the information in an unusual way. It does not store it in any particular register or accessible address. Instead, upon issuance of a "Device Status Report" command, it makes the keyboard send the information just as though someone had typed it in. **AnsiStuf** must get the information with the procedure **ReadWithoutEcho**; a normal read would make the information appear on the screen when it was transferred from the keyboard buffer.

Procedure sc_SaveCursorPosition;
Procedure sc_RestoreCursorPosition;

The names of these procedures suggest their functions. ANSI.SYS will store only one cursor position at a time, but this should be sufficient for most purposes.

Procedure sc_ClrScr;

Procedure **sc_ClrScr** clears the screen and sends the cursor to the upper left-hand corner.

Procedure sc_ClrEol (column, row: Integer);

This erases to the end of the line, beginning at the column and row indicated in its parameter list.

Procedure ReassignKey (KeyCode: Integer; Extended: Boolean; NewDefn: sc_str80);

This procedure allows transportable keys redefinition. First, set **KeyCode** to the ASCII value of the key to be redefined. If the key is not in the normal ASCII character set (that is, if it is an extended key, such as a function key or a cursor-control key) set **KeyCode** to the numeric value given in the machine's technical reference manual and set **Extended** to True. (Extended-key values for the PC also are listed in Appendix G of the IBM BASIC manual.) Then set **NewDefn** to a list of strings and/or ASCII values to be generated when ANSI.SYS intercepts **KeyCode**. Separate the items in **NewDefn** with semicolons and enclose strings in double quotation marks. Any sequence is allowed.

For example, the following line would redefine the F10 key as a DIR command followed by a carriage return:

```
ReassignKey(68,TRUE, ' "dir";13');
```

The 68 is the extended ASCII code for the F10 key. The 13 is a carriage return. To redefine a key as an extended key, precede the extended key code in the **NewDefn** string with a 0. For example, the following line would redefine the F9 key as the down-arrow key followed by the Home key:

```
ReassignKey(67,TRUE, '0;80;0;71');
```

To redefine *q* as *a,* the following line might be used:

```
ReassignKey(ord('q'),FALSE,' "a"');
```

The new key definitions will stay in effect within any program that gets its input through DOS function calls.

Procedure sc_TextMode (attribute: sc_TextAttribute);

This procedure controls monochrome attributes of subsequently written characters. The data type **sc_TextAttribute** consists of the following elements: **bold**, **underscored**, **blinking**, **ReverseVideo**, and **plain**. The following code sequence would cause all subsequently written characters to be underscored, blinking,

EARTH COMPUTERS PRESENTS:

THE BEST OF BOTH WORLDS

TURBOACCEL/286

Now you can increase the performance of your IBM PC or PC/XT up to 500% with TURBOACCEL/286, the first 80286 processor card under $1000.

We started with an 8 MHz Intel 80286. Then packed 514K of 16-bit high speed RAM on board. Made it expandable. And left room for an 80287 math chip. But there's more.

The latest VSLI technology makes TURBOACCEL/286 far superior to 8086/186 accelerator boards. Our custom CMOS chip allows the 80286 to emulate the 8088, insuring software transparency and compatiblity.

So you just plug it in, turn it on and go. No special software. No programming. No communications problems. Just pure speed: TURBOACCEL/286.

All for only $995. To order call (714) 964-5784.

CIRCLE NO. 117 ON READER SERVICE CARD

TURBOSLAVE/PC

Run over 2000 CP/M programs faster than on any other computer with TURBOSLAVE/PC, the first Z-80H coprocessor for your IBM PC.

We use Zilog's 8MHz Z-80H. Include a CP/M emulator. Add on 128K RAM with parity, two RS-232 serial ports and full documentation to bring you the fastest, most versatile and reliable board of its kind.

Then we took it a step further. Under TurboDOS (optional) up to 16 TURBOSLAVE/PC users can be linked to a single PC for true multiuser capabilities. And our open architecture and utilities make TURBOSLAVE/PC ideal for those who need a special purpose processor.

All for only $495 (OEM/ISO discounts available).

So order your TURBOSLAVE/PC by calling (714) 964-5784.

CIRCLE NO. 191 ON READER SERVICE CARD

EARTH COMPUTERS

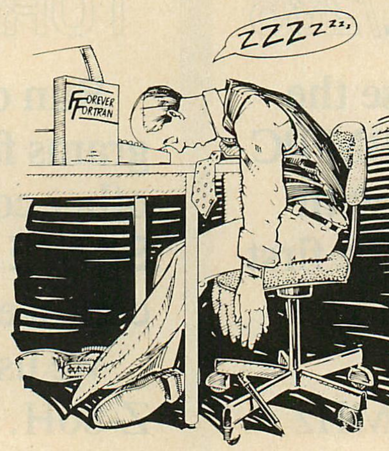
"Building Blocks For The Super Micro"

Dept. MP
P.O. Box 8067, Fountain Valley, CA 92728
TELEX: 910 997 6120 EARTH FV
FOR MORE INFORMATION AND QUANTITY DISCOUNTS
CALL (714) 964-5784

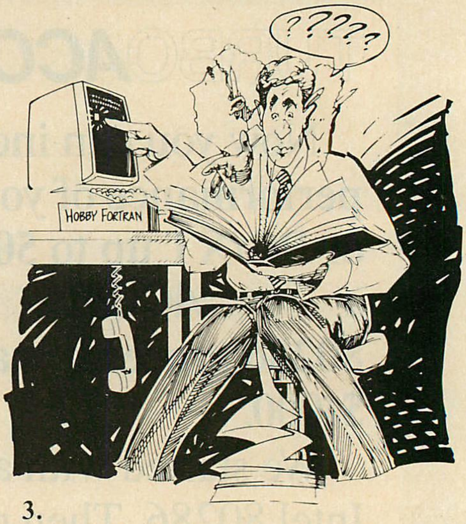
FORTRAN PROGRAMMERS

LOOK FAMILIAR?

1.



2.



3.

F77L GETS YOU OUT OF THESE SITUATIONS

1. PROGRESS IMPOSSIBLE

You've tried all the FORTRANS, but none of them allow you to complete your project. It could be the lack of key features, the overwhelming bugs, being stuck with a subset, or the inability to run large programs.

The answer to your problems is **F77L**. With **F77L** you not only get the full ANSI(X3.9—1978) Standard but additional features for flexibility in programming. We have done more than simply design a product to run all your programs, we implemented a system that allows users to reach their full programming potential. At LCS, we have been specializing in FORTRAN for over 16 years, and we believe that our commitment and dedication to FORTRAN has resulted in the finest language system available.

2. THE ENDLESS COMPILE

Your valuable time is spent waiting endlessly for your programs to compile. As your waiting increases you fall further and further behind on your FORTRAN project.

If pride in your work and meeting deadlines are important to you then you can't afford to be without **F77L** any longer. With **F77L** we have not only developed an unbelievably fast compiler but we did it without sacrificing features, diagnostics, or the speed of execution. This uncompromising attitude is evident in the quality and performance of **F77L**. Whether you are downloading from mainframes or developing on the personal computer, you can appreciate the importance of compiler speed.

3. MYSTERY MESSAGES

The messages your FORTRAN is giving don't make any sense.

F77L eliminates the confusion by explicity diagnosing errors. The compiler and run time messages describe, in plain English, the nature and location of the error. For logical program errors, we provide our unique Source On-Line Debugger (SOLD). SOLD allows you to interface with your program at the FORTRAN level and doesn't require re-compiling or re-linking. If you happen to run into an especially tough problem, you can call our technical staff for assistance.

COMING SOON:

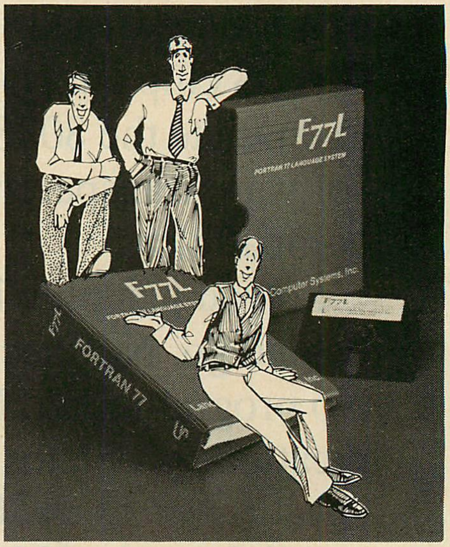
- NEW MANUAL
- LARGE ARRAYS, GREATER THAN 64K
- SOURCE ON-LINE DEBUGGER UPDATED/NEW FEATURES

CIRCLE NO. 128 ON READER SERVICE CARD

Isn't it time to say good-bye to all the shortcomings of your old FORTRAN system, and get back to enjoying programming.

F77L

"THE PROGRAMMER'S FORTRAN"



$477

Requires: PC/MS-DOS, 256K, 8087

TO ORDER OR FOR MORE INFORMATION

(213) 541-1200

Lahey Computer Systems, Inc.

31244 Palos Verdes Dr. W., Suite #243
Rancho Palos Verdes, CA 90274

and in reverse video:

sc_TextMode (underscored);
sc_TextMode (blinking);
sc_TextMode (ReverseVideo);
WriteStr (outp, 'This has all 3 attributes.');

To return to the normal display mode, execute

sc_TextMode (plain);

Procedure sc_TextColor (foreground, background: sc_color);
This procedure sets the colors to be used as the foreground and background of each subsequently written character. The data type **sc_color** is provided to make the procedure easy to read and use. Its elements are **black, red, green, yellow, blue, magenta, cyan, and white.**
Procedure sc_ScrnMode (TheMode: sc_ScreenMode);
This procedure invokes the screen width or type specified by the parameter. Note that changing the screen mode always clears the screen. The data type **sc_ScreenMode** is provided to make the procedure easy to read and use. Its elements are as follows: **BW40x25, color 40x25, BW80x25, color80x25, color320, BW320, BW640, WrapAtEndOfLine.** When **WrapAtEndOfLine** is passed to **sc_ScrnMode**, all subsequently written lines that extend past the right edge of the screen will break and continue on the next line.
Function coord (ColNum, RowNum: Byte): Integer;
The function **coord** makes it easier to work with the routines below, which treat the screen as a linear sequence of 4,000 bytes. Examples of the use of this function are sprinkled throughout the **AnsiStuf** source code.
Procedure ReadVidCh (Spot: Integer; Var TheChar: VideoMemChar);
ReadVidCh sets **TheChar** to the character and attribute at the screen position indicated by **Spot. ScrnSave** uses this procedure to store all or part of the screen so that it later can be restored with **ScrnRestore**. For example,

ReadVidCh(coord(40,10), BufCh);

will examine the screen position at column 40, row 10, and will set **BufCh** to whatever character and attribute happens to be there. It works very quickly in DMA mode, quickly enough in ROM mode, and not at all in ANSI mode—the operating system does not include a screen-reading function. The data type **VideoMemChar** is a case-variant record that allows reference to **VideoMemChar** either as an integer or as a pair of bytes, one for the character and one for its attribute. The source code for **Write**-

FIGURE 7: *RealVideoMode*

```
function RealVideoMode: sc_ScreenMode;
var rgstr: RegisterRecord;
begin {RealVideoMode}
  MonoBdInstalled := false;
  with rgstr do begin
    ah := 15; {function that reads current
               video mode}
    intr( $10, rgstr ); {performs an
                         Interrupt 10H }
    case al of
      0: RealVideoMode := BW40x25;
      1: RealVideoMode := Color40x25;
      2: RealVideoMode := BW80x25;
      3: RealVideoMode := Color80x25;
      4: RealVideoMode := Color320;
      5: RealVideoMode := BW320;
      6: RealVideoMode := BW640;
      7: begin
           RealVideoMode := BW80x25;
           MonoBdInstalled := true;
         end;
      else abort('Mode unknown to ANSI.SYS.');
    end; {case}
    sc_MaxCol := ah;
    sc_MaxRow := 25;
  end; {with rgstr}
end; {RealVideoMode}
```

The most critical part of this routine is the determination of which video adapter is installed.

At gives an example of one convenient method of setting the attribute byte.
Procedure WriteVidCh (spot: Integer; TheChar: VideoMemChar);
WriteVidCh is the converse of **ReadVidCh**, but it works the same way (though at greatly different speeds) in all three modes. **WriteVidCh** handles character-oriented screen displays much more quickly than **WriteAt** does. **WriteAt** is, however, faster for string-oriented displays.
Function RealVideoMode: sc_ScreenMode;
This function (see figure 7) uses interrupt 10H service routine 15 to determine the true current video mode (that is, **BW80x25, Color40x25,** etc.) and the true current width of the screen. For **AnsiStuf**'s purposes, however, the most critical part of the routine is the determination of which video adapter is installed (monochrome or color/graphics). **RealVideoMode** modifies a global Boolean variable named **MonoBdInstalled**, allowing **InitAnsiStuf** to initialize a global pointer (**VidPtr**) used in all DMA video routines. The step is important, because the memory on the monochrome and color/graphics display adapters is at different addresses.
Procedure WriteAt (ColNum, RowNum: Byte; TheStr: dos2str255);
WriteAt should be the most frequently called I/O routine in any program that is designed to take advantage of the **AnsiStuf** module. If the user selects ANSI

or ROM VideoMethods, procedure WriteAt moves the cursor to the screen position indicated by ColNum and RowNum and writes TheStr using any previously selected text attributes. The routine performs in exactly the same way if the user selects VideoMethod DMA, with the exception that it executes much more quickly and does not move the cursor. In order to guarantee consistent cursor placement in all different VideoMethod modes, a program should execute sc_GotoXY (NominalCol, NominalRow) following every unbroken sequence of WriteAt commands.

Procedure CursorHeight (lines: Byte);
CursorHeight lets a program control the cursor's appearance. It works only in ROM and DMA modes because neither the operating system nor ANSI.SYS include cursor-control functions. CursorHeight(0) makes the cursor invisible. CursorHeight(2) gives the cursor its normal, blinking-underscore appearance. With the monochrome display adapter, CursorHeight(13) creates a full-height cursor. CursorHeight(7) does the same with the color/graphics display adapter.

Procedure sc_ClrPart (col1, row1, col2, row2: Integer);
This procedure clears a portion of the screen. Col1 and Row1 should indicate the top left corner of the area to be cleared. Col2 and Row2 should indicate the bottom right corner of the area.

Procedure ScrnSave (Var scrn: ScreenType; col1, row1, col2, row2: Integer);
This procedure stores the appearance of part of the screen. It uses dynamic memory allocation, so that no maximum number of savable screens has to be selected at compile time. Col1 and Row1 should indicate the top left corner of the area to be stored. Col2 and Row2 should indicate the bottom right corner of the area. This routine can be used in conjunction with the sc_ClrPart and ScrnRestore procedures to implement a transportable version of the sort of removable windowing package described in "The Power of Turbo Pascal" (Michael Covington, *PC Tech Journal*, February 1985, p. 112).

Procedure ScrnRestore (scrn: ScreenType; col, row: Integer);
This procedure restores a portion of the screen stored with a previous ScrnSave command. The addition of column and row parameters in ScrnRestore allows it to restore the screen portion at a location different from the one at which it was saved or to copy from one part of the screen to another.

Procedure ReadEnvironment (name: dos2str80; Var parameter: dos2str80);
ReadEnvironment resembles the GetParameter procedure described in "Environmental Excavations" (Robert B. Stam, *PC Tech Journal*, February 1985, p. 90.) It returns the current setting of any option selected with the DOS SET command. AnsiStuf uses ReadEnvironment to determine the parameter of an environment string named VideoMethod. In addition, ReadEnvironment has other applications. For example,

```
ReadEnvironment ('comspec', YourStr)
```

will set YourStr to a string representing the drive/path location of COMMAND.COM. The command

```
ReadEnvironment ('path', YourStr)
```

sets YourStr to the operating system's current PATH setting.

One critical fact should be noted about the DOS SET command. At least as it is implemented in DOS 2.0, SET does not strip any blanks that may lead or trail the equal sign (=), and it considers these blanks significant in changing the value of an environment string. Thus, the following results can be obtained with the SET command (user input is lowercase; operating system response is uppercase):

```
A>set
COMSPEC=A:\COMMAND.COM
PATH=C:\;A:\;B:\
A>set videomethod = rom
A>set videomethod=dma
A>set
COMSPEC=A:\COMMAND.COM
PATH=C:\;A:\;B:\
VIDEOMETHOD = rom
VIDEOMETHOD=dma
```

Users should be consistent in choosing a method of setting the VideoMethod parameter; setting it once with spaces and once without will not change the first setting and will not affect the VideoMethod the program selects.

DOS2IO-3.INC

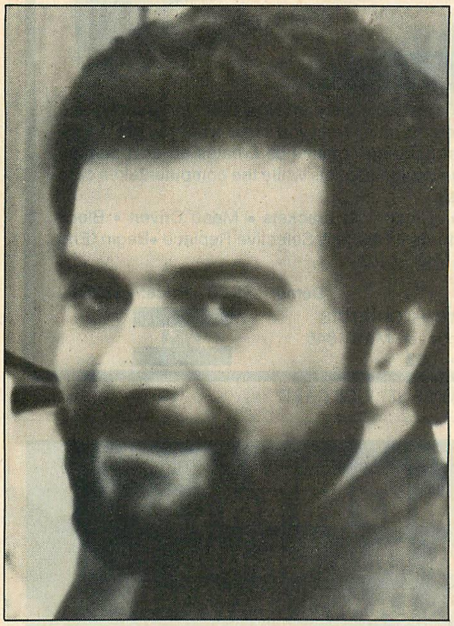
Dos2io-3.inc uses routines in Dos2io-1.inc and Dos2io-2.inc to solve three of the most commonly encountered problems in sophisticated file-handling programs: accessing disk-directory information, determining (from the volume label) which disk is presently in a drive, and copying files and transferring them between disks and subdirectories.

Although Dos2io-3.inc performs higher-level functions than either Dos2io-1.inc or Dos2io-2.inc, it tends to accomplish these functions using lower-level techniques. Dos2io-3.inc

"I Program In BetterBASIC And I Recommend It."

"It would have taken me four months in FORTRAN to write DEA's program which is a sophisticated data base and process control system that regulates all the processes used in our circuit board manufacturing etching line. But in BetterBASIC I wrote the program in a much shorter time. I use BetterBASIC and I recommend it."

Claudio Edelman · DEA Equipment · Tempe, Arizona



640K Now you can use the full memory of your PC to develop large programs.
STRUCTURED Create well organized programs using procedures and functions that are easily identified and understood and completely reusable in future programs.
MODULAR Use procedures and functions grouped together to form "library modules" which are then available to you or anyone else for future use.
EXTENSIBLE Create your own BetterBASIC modules which contain BetterBASIC extensions. This feature coupled with the easy-to-use Assembly Language support, makes this an ideal OEM language.
INTERACTIVE BetterBASIC acts like an interpreter because it responds to the users' commands in an immediate mode. However each statement is actually compiled as it is entered.
COMPILED Each line of the program is compiled as it is entered into the computer's memory rather than interpreted at runtime.
RUNTIME SYSTEM The optional Runtime System generates stand alone EXE. files allowing for the distribution of products written in BetterBASIC with no royalties.
SUPPORTS Windows, Graphics, DOS and BIOS ROM calls, Chaining, Overlays, Local and Global Variables, Recursion . . . and more.
BetterBASIC Runs on IBM PC, XT, AT and all IBM-compatibles. Ask your local dealer for BetterBASIC or call 1-800-225-5800 in Canada call 416-469-5244. Also available for the Tandy 1000, 1200, AND 2000 AT Tandy/Radio Shack stores.

Summit Software Technology, Inc.™
P.O. Box 99, Babson Park
Wellesley, MA 02157

PRICES:

BetterBASIC	$199
8087/80287 Math Module	$99
BTrieve™ Interface	$99
Runtime System	$250
Sample Disk with Tutorial	$10

Better BASIC.

Because It's The Best.

MasterCard, Visa, Checks, Money Order, C.O.D. accepted and P.O. on approval.

(If you're using BetterBASIC and would like to be featured in one of our ads, please write to the Director of Advertising at Summit.)

CIRCLE NO. 169 ON READER SERVICE CARD

must concern itself with whether or not many of its numbers have their most significant bytes first or last, in part because the operating system is inconsistent in the manner in which it stores date and time information.

The services that are provided in Dos2io-3.inc are described below.

Procedure IntegerToDate (**TheInt**: Integer;
 Var **month, day, year**: Integer);
Function DateToInteger (**month, day,**
 year: Integer): Integer;
Procedure IntegerToTime (**TheInteger**: Integer; Var **hours, minutes, seconds**:
 Integer);
Function TimeToInteger (**hours, minutes,**
 seconds: Integer): Integer;

These four procedures transfer dates and times into and out of the dense format in which the operating system stores date and time information—each value is stored in only two bytes. (Refer to the DOS 2.0 manual, p. C-5; also see "Packing the Date and Time," Patrick Finan, *PC Tech Journal*, January 1985, p. 46 for more information.) If it is necessary to know precisely how these routines mesh with those listed below, it is helpful to note that the operating system considers the byte order used in these routines (and in most other parts of **Dos2io**) to be reversed. When date and time information is retrieved from the disk transfer area (DTA), the high and low bytes are in the opposite order. **Dos2io** swaps bytes that come from the DTA, not bytes that come from the registers (as suggested by the DOS manual), because the byte order used in register transfers is the same as the byte order Turbo Pascal normally uses.

Procedure SetFileDateAndTime
 (**FileHandle, month, day, year, hours,**
 minutes, seconds: Integer);
Procedure GetFileDateAndTime
 (**FileHandle**: Integer; **Var month, day,**
 year, hours, minutes, seconds:
 Integer);

The names of these two procedures suggest their functions; they set and retrieve date and time information for the file identified by **FileHandle**. In **SetFileDateAndTime** the year may be specified with any integer between 80 and 199 or 1980 and 2199. **GetFileDateAndTime** returns the year as an integer in the range 80 to 199.

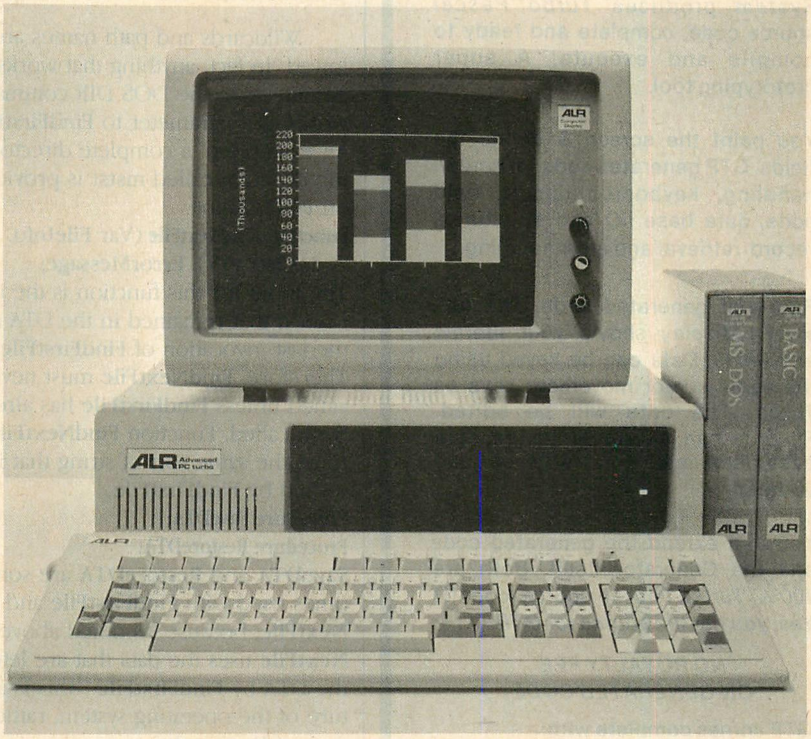
Procedure SetTime (**hour, minute,**
 second, hundredth: Integer);
Procedure GetTime (Var **hour, minute,**
 second, hundredth: Integer);

These set and retrieve the time maintained by the operating system.

Function FindFirstFile (**FileName**:
 dos2str80; Var **FileInfo**: **dos2str255**):
 ErrorMessage;

DEFINING THE IDEAL PC

com • pat • i • ble/kəm-pat-ˊə-bəl/1: capable of forming a homogeneous mixture that neither separates nor is altered by electronic interaction 2: capable of operating thousands of programs utilizing the technology of the 8088-2 microprocessor 3: having the ability to run IBM PC and IBM PC/XT programs 4: having the option of exchanging data with the 1.2 Mb IBM PC AT floppy



com • pet • i • tive/kəm-pet-ˊət-iv/ 1: inclined, able, or suited to compete 2: competing at **60% faster** with 8 MHz/4.7 MHz

com • plete/kəm-plet′/1: having all necessary parts, elements or steps 2: leaving little need for add-ons 3: having **20 megabytes** storage and **640k** of memory 4: including Clock calendar, printer and serial ports 5: an enchanced keyboard with separate numeric and curser controls as standard equipment.

con • fi • dence/kan-ˊfəd-ən(t)s/1: the quality or state of being certain 2: the assurance of a THIRTY-DAY MONEY BACK GUARANTEE 3: having a full ONE YEAR WARRANTY on an entire system

con • clu • sion/kan-ˊklu-zhan/1: to reach as a logically necessary end by reasoning 2: ideal PC, ALR Advance PC turbo at $1,995 a compatible, competitive, complete system plus the confidence of the ALR guarantee.

con • tact/kan′/takt/1: an establishing of communication with ADVANCED LOGIC RESEARCH, INC. by dialing **(714) 832-7808** on any standard telephone.

IBM PC. IBM PC/XT. IBM PC AT trademarks of INTERNATIONAL BUSINESS MACHINES CORP.

ALR

Advanced Logic Research, Inc.

15455 Red Hill Ave., Suite B, Tustin, CA 92680

(714) 832-7808

CIRCLE NO. 104 ON READER SERVICE CARD

This function retrieves directory information for the first file that matches the directory search request in **FileName**. Pass a directory search string (for example, 'b:*.*') to **FindFirstFile** through its first parameter, and its second parameter will return information about the first matching file in a string of the following form:

ARTICLE5.TXT 42832 8-26-84 4:29p

Wildcards and path names are allowed. In fact, anything that works as a parameter to the DOS DIR command works as a parameter to **FindFirstFile**. An example of a complete directory-listing program called **mstst** is provided on PCTECHline.

Function FindNextFile (Var **FileInfo**: **dos2str255**): **ErrorMessage**;

The input for this function is the information that remained in the DTA after the last invocation of **FindFirstFile**. Therefore, **FindNextFile** must never be called unless **FindFirstFile** has already been called. Function **FindNextFile** returns the same sort of string that is returned by **FindFirstFile**.

Procedure SaveDTA;
Procedure RestoreDTA;

SaveDTA and **RestoreDTA** are sometimes used with **FindFirstFile** and **FindNextFile**, because, as noted above, **FindNextFile** uses the data that are left in the DTA by **FindFirstFile**—this is a feature of the operating system, rather than of **Dos2io**. If a program needs to list files from one directory while in the middle of listing those from another directory, it must save the first DTA so that it can return to it when the files from the second directory have been listed. Futhermore, if a program needs to list two or more additional directories before returning to the first one or if it needs to list two or more directories concurrently, a new DTA storage scheme will have to be devised.

A dynamically allocated stack of DTAs could be created or a set number of DTAs could be allocated at compile time and the program could switch between them using essentially the technique illustrated in the source code for **SaveDTA** and **RestoreDTA**.

Function VolumeLabel (**TheDrive**: Char): **dos2str80**;

This function returns the volume label of the disk in the drive that is identified in its parameter. If the disk has no label, it returns the message "unlabelled vol." The source code for function **VolumeLabel** exposes an interesting difference between the directory-searching function calls for DOS 1.0 and those for DOS 2.0. The documentation for function **$4E** (find first matching file), which is used in **FindFirstFile** and **FindNextFile**, suggests that it provides the same services as function **$11** (search for the first entry). The documentation does not mention that function **$4E** does not consider the volume label a file.

Function FreeDiskSpace (**DriveLetter**: Char; Var **FreeBytes**: Real): **ErrorMessage**;

FreeDiskSpace returns the number of available bytes on the drive identified by **DriveLetter**. **DriveLetter** can either be upper- or lowercase.

Function CopyFile (**OldHandle**: Integer; **NewFileName**: **dos2str80**): **ErrorMessage**;

CopyFile works much as the DOS COPY command does. First it determines the length of the file to be copied, represented by **OldHandle**. Then it uses **FreeDiskSpace** to determine whether there is room to copy the file to the destination disk optionally indicated in **NewFileName**. In making this determination, **CopyFile** takes into account the number of bytes that will be freed if an existing file on the new disk is overwritten; COPY does not seem to do this. If there is enough room, **CopyFile** creates the new file on the drive and path indicated and copies from the old file to the new in chunks of, at most, 64KB.

Turbo Pascal finally has made Pascal competitive with BASIC—which, after all, has been provided free with the vast majority of all microcomputers since Apple and Radio Shack began the tradition in the late 1970s. One of BASIC's hallmarks has been its rich repertoire of I/O statements. Pascal has been criticized for lacking I/O—yet its extensibility allows whatever I/O features are needed to be created with little or no speed penalty. These routines should allow Turbo Pascal to perform any I/O that can be performed with BASIC—and more.

Since this article was written, Borland International has released Turbo Pascal 3.0, which incorporates some of the file-handle and path-name support features present in Dos2io. Nevertheless, the source code for Dos2io offers some fascinating insights into how low-level file functions and procedures may be constructed in what is usually seen as a high-level language. —*JD*

Cole Brecheen is a former editor-in-chief of the Harvard Computer-aided Legal Instruction Project at Harvard Law School. He is a clerk for the Oregon state court of appeals.

Everything You Always Wanted To Know About

APPLICATION DEVELOPMENT DATABASES...

DATA

Just What Is DataFlex, Anyway?

The easiest thing to say is that DataFlex is an application development system *like* d-BASE. However, that wouldn't be a fair statement about either product because DataFlex is not only faster, more powerful and easier to use than d-BASE, but also supports TRUE multi-user transaction processing with complete data integrity. Applications developed with DataFlex can also be run UNCHANGED on a wide selection of 8 and 16-bit operating systems and LANs.

More Powerful and Easier To Use?

Absolutely. DataFlex uses "image formatting" to quickly and efficiently develop input screens and report formats for your application. All you have to do is make an "image" of your screen or form using any ASCII text editor. AUTODEF, DataFlex's file definition utility, automatically generates an error-free, ready to compile data entry program, creates the necessary data and key index files, and makes a data dictionary entry describing each field, its length, type and format. With DataFlex, there's no need for you to go back and provide tedious definitions of the length, format and data type of your windows. And while we're on the subject of text editors, you should know that DataFlex is also available with a full function word processing option that's operationally equivalent to and compatible with MicroPro's WordStar + Mailmerge.

What About A Procedural Language?

DataFlex has a powerful procedural language that combines the best features of Pascal, BASIC and RPG. It has over 125 commands in the following categories:

Argument Processing	Indicators
Console I/0	Key Procedures
Control	Multi-User Functions
Database Commands	Reporting
Definition	Sequential I/O
Data Entry	String Operations
Forms Processing	Structural Control
	System Commands

DataFlex allows programs of up to 2,750 command lines incorporating up to 255 screen images! This enables you to design systems far more complex and sophisticated than possible with some other database programs. For many applications, however, knowledge of the procedural language is not necessary since DataFlex's powerful AUTODEF and QUERY functions automatically generate the source code for handling data entry and reporting. Data checking and formatting commands and error traps can easily be added to the source code before compiling so your applications will run smoothly with minimal possibility of operator error.

How Can DataFlex Be So Fast?

DataFlex utilizes a multi-keyed B+ ISAM structure which updates indexes on-line each time data is entered, deleted or edited. Since all data is instantly available for recall, time consuming key sorts and batch index reorganizations are not necessary. With DataFlex, you'll never again have to wonder whether or not your computer is really working or "hung-up" somewhere in the middle of a sort. DataFlex reports appear on your screen or printer as quickly as the data can be read from your disk. Data input is also speeded by DataFlex's FlexKeys™: single keystroke commands that perform record finding, saving and editing functions. There's even a HELP FlexKey that can summon instructions or explanations pertinent to your application.

Will DataFlex Exchange Data With Other Programs So I Don't Have To Start All Over From Scratch?

Yes! Import and export of data is a snap. DataFlex can read or output either comma or carriage return/linefeed delimited files. Conversion of d-BASE data files is described in detail in our comprehensive user's manual.

The Application Development Database For Single-User, Multi-User And Local Area Networks

FLEX™

What's DataFlex Have For Those Of Us With Heavy Reporting Requirements?

Output for reports, labels, preprinted forms and files is handled by a powerful DataFlex macro command called REPORT. It consists of an integrated set of selectable predefined output routines that can be chosen as needed. These routines are used to "fill-in" an output "image" and then send it to the device of your choice. You can write the report command file yourself or it can be automatically generated by DataFlex's QUERY function. Even complex multi-file reports can be generated through QUERY. All you do, using the arrow keys or a mouse, is "point-and-shoot" at the data you want to see! QUERY then automatically writes error-free source code and allows you to save it as an ASCII file that you can then customize, compile and run. Output can be sent directly to your printer or CRT, or saved as a comma or carriage return/linefeed delimited ASCII file for later use by DataFlex or some other program. The speed with which QUERY performs its source code generation function is something that you have to see for yourself to fully appreciate.

SUPPORTED OPERATING SYSTEMS AND NETWORKS

IBM PCDOS 1.x, 2.x, 3.x
MSDOS 1.x, 2.x, 3.x
IBM "AT" XENIX
IBM PC NETWORK
CP/M, CP/M-86
Concurrent CP/M-86
Concurrent DOS
MP/M-86
Novell NetWare
TurboDos
Corvus w/IBM PC
Molecular N/Star
TeleVideo Infoshare
3Com EtherShare
PC-Net
NorthStar Dimension
Action DPC/OS
DMS Hi-Net
Alloy Engineering RTNX

SPECIFICATIONS

Environment:
- 8080, Z80, 8088, 8086, 80186, 80286

Requirements:
- 52K TPA (8-bit)
- 256K TPA (16-bit)
- CRT w/cursor addressing
- 600K disk storage

Capabilities:
- 255 Database files
- No limit on number of open files (16-bit)
- 9 6-segment indexes per file (16-bit)
- 16K Bytes per record
- 255 Fields per record
- 16,777,215 Records/file
- 2,750 Program lines per configuration
- 255 Screens per configuration
- 2,900 Windows per configuration
- 32,000 Variables
- 127 Indicators
- 9 Global break point levels
- 18 Terminal independent function keys

Does DataFlex Work On MULTI-USER Systems and LANs?

Yes! DataFlex, unlike most other products, supports TRUE multi-user processing. With DataFlex, no user is ever locked out of a file or denied access to a record. Every user can, at any time, access, read and even change any record in the database while maintaining absolute data integrity!

How Easy Is It For The "End User" To Deal With DataFlex Applications?

DataFlex includes an elegant menu system which totally insulates the end-user from the computer's operating system. Each menu screen supports up to nine prompted actions each, including chaining to "sub-menus" and DataFlex programs, and the execution of system commands and other programs. A pre-programmed "help-screen" is included to provide operator assistance on selecting items from the menu. Password security can be established for each menu action to prohibit unauthorized file access, and the passwords (or even the entire menu) can easily be changed at any time by programmers with access to DataFlex's MENUDEF utility.

With Some Programs, I've Had Performance Go Right Down The Tube When There's More Than A Thousand Or So Records. How's DataFlex Perform With Large Databases?

Large, complex databases are exactly what DataFlex was designed for, and its performance in this environment is impressive. In benchmark tests on a Wang PC with a 36,000 record database of 128 byte records, DataFlex was able to find a record via a 41 byte key and display it to the screen in .8 seconds! This high level of performance extends to multi-file operations as well, where the PC version of DataFlex puts no limit on the number of open database files. As many as 255 database files can be maintained by DataFlex, the size of each limited only by your operating system and DataFlex's 16.7 million record per file "limit."

DataAccess Corp., 8525 S.W. 129th Terrace, Miami, FL 33156 (305) 238-0012 TELEX 469021 DATA ACCESS CI

CIRCLE NO. 118 ON READER SERVICE CARD

TK!Solver Equates

With TK!Solver, the user can create a model of almost any phenomenon or system that can be described mathematically.

VICTOR E. WRIGHT

TK!Solver is most often described as a tool kit (thus the TK) or an equation solver. While these two simple explanations are not incorrect, anyone who has spent time working with TK!Solver knows that this program is much more. Its facilities can, for example, be used to solve real-world engineering problems.

Software Arts' TK!Solver *is* an equation solver. The user enters an equation, providing input values, and TK solves the equation, providing output values for the unknown variables. In fact, systems of simultaneous equations can be entered, with the appropriate number of knowns, and TK will solve the systems of equations for several unknowns. TK is a modeling tool, specifically, a mathematical modeling tool. It allows the user to create a model of almost any phenomenon or system that can be described mathematically. TK also functions as an electronic version of a technique for organizing the parts of a mathematical model—the engineer's or scientist's notebook.

PHOTOGRAPH • MARC DAVID COHEN

The concept of using a machine to duplicate manual techniques may not have originated with microcomputers, but Software Arts certainly made the concept popular with VisiCalc (an electronic spreadsheet) and it continues the idea with TK!Solver. TK is not a direct descendant of VisiCalc, but the two products were created from a similar perspective—they both use the computer to enhance proven techniques.

Modeling serves similar purposes in many fields. Engineers, scientists, and economists all study models because certain phenomena cannot be contained in a laboratory or studied at a desk. Sometimes the subject exists only in conceptual form. An economist cannot directly observe the workings of the economy of even a small town, so he constructs a mathematical model. Similarly, the designer of supersonic aircraft cannot observe the forces that act on the vehicle in flight. He builds several types of models, among them mathematical models of various processes and physical scale models for wind tunnel testing. These models offer the observer a close look at otherwise intangible real-life phenomena.

Most systems in the physical sciences can be described mathematically. Mathematical models comprise equations, variables, sets of values, and various presentations of the values. To be of practical use, most models must have units associated with the variables. Tables and graphs are generally included as an aid in presentation.

Efficient modeling requires tools—rigorous techniques for solving equations and systems of equations, for keeping track of units, for summarizing results in tabular or graphic form. Some systems of equations can be solved by direct substitution whereas others require iteration. Control systems engineers use Laplace transforms to solve differential equations. Engineering abounds with numerical techniques.

TK's ORGANIZATION

TK!Solver resembles a spreadsheet in that the screen is divided into fields, but the similarity ends there. The fields of a spreadsheet are identical—the user can enter any legal entry in any field; TK's fields are specialized and are grouped by function.

TK accepts and executes commands. Like VisiCalc, TK normally expects the user to enter data and it highlights the field in which data will be accepted. The manner in which commands are issued to TK resembles the VisiCalc syntax: all but five of TK's commands begin with a slash (/) character, and many of its commands are exact VisiCalc duplicates.

FIGURE 1: *Sample Variable Sheet*

```
(1n) Name:                                                      85 /!

=================== VARIABLE SHEET ====================================
St Input    Name    Output    Unit       Comment
-- -----    ----    ------    ----       -------
                                         PIPE SIZING MODEL
                                         *****************
L           dP      .64Ø93Ø61 fth2o      pressure drop in L feet of pipe
            f       .Ø17245Ø4            friction factor, dimensionless
L  1        L                 ft         length of pipe
            rho     62.4      lbm/ft^3   fluid density
   5        V                 ft/s       fluid velocity
L           D       .28584123 in         inside diameter of pipe
            g       32.174    ft*lbm/lb  acceleration of gravity
L  1        Q                 gal/min    volume flow rate of fluid in pipe
L           Re      9794.45Ø2            Reynolds number
            v       .ØØØØ1216 ft^2/s     kinematic viscosity
            epsilon .ØØØ15    ft         absolute roughness
L           Ds4Ø    .364      in         standard sch 4Ø pipe id
L           dPs     .191394Ø8 fth2o      pressure drop in sch4Ø pipe
L           Vs      3.Ø833Ø86 ft/s       velocity in sch 4Ø pipe
L           Res     7691.3674            Reynolds number in sch 4Ø pipe
L           fs      .Ø392Ø827            friction factor in sch 4Ø pipe
L           Dnom    'one_qtr             nominal pipe size
```

The fields contained in the variable sheet provide virtually all necessary information about a variable in a rule (equation). The status field is used to pass important current status information about the variable; for example, it can inform the user of errors. The variable sheet above is for the pipe-sizing model.

When the slash is typed as the first character in an entry, TK displays a prompt line at the top of the screen, listing the first letter of each command that can legally follow the slash. Users unfamiliar with VisiCalc (or TK) may require a reference chart.

Like the engineer's notebook, TK!Solver is organized into sheets. The user constructs a TK model by making entries in the various sheets. The sheet organization is itself a model—the user's model of the program. TK is not actually organized into sheets in the computer's memory, but the sheet model is a natural way to think of the program's organization.

One of TK's commands, the window command, offers another way to think about TK models. Just as the screen display of a spreadsheet is a window on the entire worksheet contained in memory, TK's sheets are a series of windows on the entire model. The window command allows the user to change the display from a single sheet to two sheets, or vice versa. Although another command, the select command, is used to change the sheet displayed in a portion of the screen, the window command suggests that the screen is a window into a model too large to be displayed in its entirety.

The two basic sheets are the rule sheet and the variable sheet. The user enters equations in the rule sheet, just as they appear in textbooks or handbooks. TK's equations, or rules, are *true* equations in the same sense that textbook equations are; a TK rule says that the terms on the left side of the equal sign, when evaluated, are equal to the terms on the right side, when evaluated. (This use of the equal sign is, of course, different from its use in BASIC, for example, where it signifies assignment, not a condition of equality.)

The variable sheet is the second necessary ingredient of a TK model. When the user enters a rule, or a series of rules, into a TK model, the program examines (parses) the rule, and extracts the names of the variables for display on the variable sheet. Again, this display follows the textbook method of presenting mathematical models, except that TK places the variable sheet on top and the rule sheet on the bottom (most texts list the equations on top with the variables listed below).

Figure 1 shows the variable sheet, which contains several fields besides the one that lists variable names. It is more like a spreadsheet in that it has six columns of fields. The fields are (from left): status, input, name, output, units, and comment. These fields provide virtually all necessary information about a variable in an equation. The name, units, and comment fields provide data similar to that found in the textbook presentation of an equation or formula. The function of the input and output fields is obvious; the status field may require explanation.

Status fields are not unique to the variable sheet; the rule sheet has status fields and rule fields. In the variable sheet, TK uses the status field to inform the user of errors. The field can be used to associate variables with lists, to associate a variable with a guess value, and to move a value from the output field to the input field. In the rule sheet, TK uses the status field to signal whether the rule is satisfied, unsatisfied, or has caused an error.

Each row in the variable sheet is supported by a variable subsheet that contains the information displayed in the variable sheet and some additional items. A variable's subsheet can be examined by placing the highlight on the desired row and typing >, which is the dive command. To return to the previous display (the variable sheet), type <, the return command.

If TK included only the direct solver, *it would be a powerful program; however, its second method, the* iterative solver *makes possible the solution of real-world engineering problems.*

To create a model TK can solve, the user makes entries in the rule and variable sheets. The basic model is constructed by entering the rules in the rule sheet. TK automatically makes an entry in the name field for each variable in the rule sheet. When the rules are entered, the input values can be entered into the input field of the variable sheet, then the model can be solved. TK can be instructed to solve the model either by issuing the action command, !, from the rule sheet or the variable sheet or by issuing the solve command, /!, from any sheet.

TK!Solver incorporates two methods of solving models. The first and preferred method is the *direct solver*, which works in much the same way as manual solutions. It scans the model looking for a rule with only one unknown, solves that rule, and replaces all occurrences of the unknown with the newly found value. Then it scans the model again, looking for another rule with only one unknown. The process continues until all variables are bound to values. If the user enters a model that consists of simultaneous equations, one with only one unknown, another with two, another with three, and so on, the direct solver will reach a solution quickly. (Other conditions must be met in order to use the direct solver; they are discussed later.)

However, most serious engineering problems are not so well organized. Typically, a set of simultaneous equations will contain no rules that have only one unknown. Although the user may be able to manipulate the rules in order to obtain a set of equations that TK can handle with the direct solver, that is usually unnecessary.

If TK!Solver included only the direct solver, it would be an interesting program and more powerful than most other equation-solving products. TK's second solving method, the *iterative solver*, however, makes feasible the use of TK for the solution of real-world engineering problems that typically require numerical methods.

The iterative solver is triggered by the presence of a guess value in the model's input, indicated by a G in the status field of a variable or by the appearance of a variable more than once in a single rule. The product's manual does not describe the iterative solver in detail, although it does provide insight into the notion of iteration. Other literature confirms that the TK!Solver method is based on a modified Newton-Raphson procedure.

MANEUVERING VALUES

TK's list processing features help make the product handy for engineering use. They permit creating lists of values, either numeric or symbolic; the user types successive values into a list subsheet. Then, that list of values can be associated with a variable, usually an input variable, by entering **L** in the variable sheet status field for that variable. Finally, the list processor is invoked with the list command, /L. The list command offers four options—solve, !; solve block, **B**; put, **P**; and get, **G**.

Issuing the command /L! will cause TK to solve the model one time for each value in the longest input variable list, producing lists of output values for each output variable associated with a list; as is the case with input variables, an output variable is associated with a list by entering **L** in its variable sheet status field. The list processor is invaluable for solving repetitive problems, such as sizing pipe for fluid flow and sizing electrical conductors.

Lists can be displayed as list subsheets in tabular form. A list subsheet contains the units associated with the list, a comment field, a column of element numbers, and a column of values. The list subsheets in the model are summarized in the list sheet.

TK includes 34 built-in functions that can be included in rules or in expressions entered in input fields. Twenty-seven of them are functions that might be expected—trig functions, MIN, MAX, LOG, and so on. The other seven are list related; they include COUNT, to count the number of elements in a list; APPLY, to invoke user functions; DOT, to form the dot product of two lists; ELEMENT, to retrieve specific elements of lists; GIVEN, to protect the model against missing input values; POLY, to evaluate polynomials; and STEP, to branch, depending on the relative values of two arguments.

TK also permits the user to define his own functions. However, TK's user-defined functions are not the same as user-defined functions in programming languages. In most languages, a function can be defined as a series of procedural statements that execute and return a value to the calling statement.

TK's user function is a pair of lists—a domain list, and a range list (that is, an independent variable list and a dependent variable list). The function is invoked with an argument from the domain list; it returns the corresponding value from the range list.

In addition, the user may specify the mapping (the way values are interpolated) between the two lists. Three mapping options are available: table, step, and linear. In table and linear mapping, TK returns a value from the range list if the function argument is unknown. It would be helpful to form user functions from combinations of other functions or procedures, but TK's user functions are helpful as they are; depending upon the mapping, user functions can serve as look-up tables, nomographs, and approximations to curvilinear functions.

It is helpful to create user functions that describe empirical relationships or that are only piecewise continuous. TK provides two handy ways to assist in the filling of a list function. The user can enter the first and last elements in a list subsheet and issue the action command, !, from that sheet. TK then fills the list with intermediate values separated by a constant interval. The user can create a model to fill the lists of a user function, and then save the function to disk for use in other models. User functions may be called directly or via the APPLY function.

As with list subsheets, user functions are described in detail on user function subsheets and in summary form on the user function sheet. The range and domain lists are contained in list subsheets, as are any other lists.

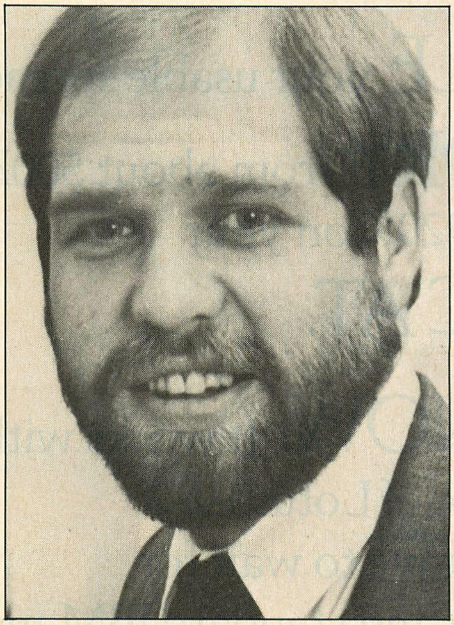
The user can examine a list at any time; he can, in fact, examine two lists at one time by displaying the list sheet in both windows, and diving to a different list in each of the two windows, however, TK provides an even better facility for displaying lists.

The first option displays several lists in tabular form via the table sheet, which is a set-up sheet. The user begins the set-up procedure by entering the names of the lists to be included in the table; he then makes entries to define the format of a single table, which can display several lists. Tables in the model are defined one at a time. Once the table is defined using the table sheet, the user issues the action command from the table sheet, and TK displays the table on the screen or on the printer.

"I Program In BetterBASIC And I Recommend It."

"I use BetterBASIC over C and PASCAL because it's more versatile and convenient. My department acts as an in-house consultant at Fannie Mae. We evaluate software for the entire organization. When the need arises for a programming language, I recommend BetterBASIC."

Lee Beckley · Information Center Analyst · Fannie Mae · Washington, D.C.



640K Now you can use the full memory of your PC to develop large programs.
STRUCTURED Create well organized programs using procedures and functions that are easily identified and understood and completely reusable in future programs.
MODULAR Use procedures and functions grouped together to form "library modules" which are then available to you or anyone else for future use.
EXTENSIBLE Create your own BetterBASIC modules which contain BetterBASIC extensions. This feature coupled with the easy-to-use Assembly Language support, makes this an ideal OEM language.
INTERACTIVE BetterBASIC acts like an interpreter because it responds to the users' commands in an immediate mode. However each statement is actually compiled as it is entered.
COMPILED Each line of the program is compiled as it is entered into the computer's memory rather than interpreted at runtime.
RUNTIME SYSTEM The optional Runtime System generates stand alone EXE. files allowing for the distribution of products written in BetterBASIC with no royalties.
SUPPORTS Windows, Graphics, DOS and BIOS ROM calls, Chaining, Overlays, Local and Global Variables, Recursion . . . and more.
BetterBASIC Runs on IBM PC, XT, AT and all IBM-compatibles. Ask your local dealer for BetterBASIC or call 1-800-225-5800 in Canada call 416-469-5244. Also available for the Tandy 1000, 1200, AND 2000 AT Tandy/Radio Shack stores.

Summit Software Technology, Inc.™
P.O. Box 99, Babson Park
Wellesley, MA 02157

PRICES:

BetterBASIC	$199
8087/80287 Math Module	$99
BTrieve™ Interface	$99
Runtime System	$250
Sample Disk with Tutorial	$10

Better BASIC.

Because It's The Best.

CIRCLE NO. 195 ON READER SERVICE CARD

MasterCard, Visa, Checks, Money Order, C.O.D. accepted and P.O. on approval.

(If you're using BetterBASIC and would like to be featured in one of our ads, please write to the Director of Advertising at Summit.)

The table sheet permits moderate control over the display of the table—for example, whether the table is vertical or horizontal. A vertical table displays the elements of a list in a column; a horizontal table displays them in a row. The user also can specify the width of each column (or row) and the first element of the list to display. A title (up to 200 characters) and column (or row) headings may be added. Finally, TK can be instructed to display the table on the screen or the printer.

With a second option, lists may be displayed in graphic form, using the plot sheet (TK refers to graphs as *plots*). Defining a plot is similar to defining a table, by entering the names of the lists to be plotted and the character to be used as the symbol for a data point. Since TK is designed to work on any character printer, plots consist of the data point symbols only. No interconnecting line segments exist between the points. Fairly smooth curves can be produced if the printer can be set to a very fine horizontal pitch, but some stair stepping is inevitable.

The plots can be enhanced with the addition of a title and labels for the axes. Almost any printable character may be used as the data point symbol, but command characters (/, !, ?, ;, :) must be entered with the edit function after being defined as some other character. To display a table or plot, the user positions the highlight in the table or plot sheet respectively and issues the action command. TK sends the display to the screen or the printer depending on the selection that has been made.

Although not tested for this review, TK's table sheet and plot sheet both allow a maximum 32,000 lists to be named in the definition of tables and plots. Nearly all of the sheets allow 32,000 entries in fields that are not single entry. A model size is more likely to be limited by the amount of memory available than by the program.

HANDLING UNITS

For anyone who has ever felt plagued by handling units, TK offers great relief. Units must be handled accurately in engineering problems. However, with TK!Solver, the user need worry about getting them correct only once—when the model is first set up. The program takes over from there.

Units appear in several places in a TK model. First, each variable in the variable sheet has a units field. The user can dive to the variable subsheet for any variable and find two entries for units—calculation and display—associated with the variable. Lists are similar: the list sheet has a single units column, and each list subsheet has two entries for units—storage and display. Finally, the unit sheet itself contains a list of conversion factors.

The rules that are first entered in creating a model should treat units consistently. TK knows nothing about the meaning of a model, only about ways to solve it. If the user constructs a model with inconsistent units, TK will solve it in ignorant bliss and issue a meaningless answer. Models can be solved without entries into the units field, but it is better to define a set of units and conversion factors that relate to the units and let TK do the bookkeeping.

The simplest way to enter units is to enter them in the unit field in the variable sheet. These initial entries become both the calculation units and the display units. TK makes calculations using calculation units and displays results in display units. TK has no understanding of units, only an ability to convert from one to another, once defined; the user must specify conversion factors if

he wants to display a model's results in units other than calculation units.

Display units can be changed at any time by typing in a new entry on the variable sheet: the entry in the unit field is the display unit. To change the calculation unit of a variable, the user must dive to the variable subsheet, and change the entry in the calculation unit field. TK assumes that the first set of calculation units is consistent. When the display unit of a variable is changed to one that is not listed in the unit sheet, TK will signal that the model is inconsistent by displaying a question mark beside the value. TK can apply any conversion rule in either direction, and it can apply two rules in conjunction to perform an implicit conversion. There is no need to enter every possible combination of conversion factors.

The variables in a TK!Solver model can be assigned either a numeric or a symbolic value. Numeric values are simply real numbers. TK cannot handle complex or imaginary numbers. Numeric values may be entered in scientific, exponential, or decimal form, and TK will display them in the most appropriate form. TK also permits entering expressions that evaluate to real numbers in any field that normally accepts a real number as input.

Symbolic values are character strings. They must begin with the single quote mark, ', and may not contain blanks or numeric operators. The user can create lists of symbolic values, functions that return symbolic values, and rules that invoke those functions. This capability permits the creation of models that produce output in familiar terms, such as '6''_Sch_40_pipe, instead of .5054166667 (the inside diameter of 6-inch Schedule 40 pipe, in feet).

CONTROLLING AND STORING

The global sheet, which is divided into three sections, is like a control sheet. The first section controls whether the program enters the variable names appearing in the rules into the variable sheet automatically. Some users opt to leave this field set for automatic insertion by the program.

The second section provides some control over the operation of the iterative solver. For example, TK normally displays the values of all the variables for each iteration the program makes, leaving the final set of values on the screen when the solution is complete. These intermediate solutions are very useful when testing a model, but displaying them slows the program's operation. Once a model performs correctly, the program can be set to display only the final set of values, thereby helping it to reach a solution faster.

Normally TK will attempt 10 iterations before stopping. This number can be changed from the global sheet, which may be desirable for models that converge slowly on solutions.

The second section also determines the accuracy of solutions. TK compares values to determine when it has reached a solution to a model. It normally considers two values to be equal if they are within .000001. This value can be changed by adjusting two other values, the comparison tolerance and the typical value, both of which are contained in fields in this section.

The third section of the global sheet controls the appearance of the printed output of the model. The user can control the number of lines per page, the number of characters in a line, margins, and the number of lines of printed output. He also can change the name of the output device if he wants to send ASCII output to a disk file instead of to a parallel or serial port. This section contains the field that sends the set-up string to the printer.

The model storage and retrieval capacity is an important feature of TK!Solver. TK's command syntax for

performing storage and retrieval is again similar to that of VisiCalc. The basic command for storage is /S. The options that can follow the /S command are load file, save model, save variables, save unit sheet, save functions, DIF storage, and delete file.

These storage options permit the creation of one or more files, each containing only a unit sheet—one for structural units, one for electrical units, one for heat transfer units, and so on. With these files as a basis, the user can develop a library of models, incorporating the appropriate unit file with a single command. Similarly, the user can store user functions as he develops them, and load them into later models. The variable sheet and subsheets also may be stored and loaded separately.

A new model can be created from other complete model files. When a TK model is loaded, in complete or partial form, the program does not overwrite the current model. It adds information from the new rule, unit, table, and plot sheets to the bottom of the respective sheets, even if doing so duplicates entries in the current model.

However, the program does not duplicate information located in the variable, list, and function sheets. Values in the current model are overwritten by the values of the new file, in the case of duplicate variable names.

SOME PARAMETERS

TK's ability to communicate with other programs is limited. It supports the DIF convention, so lists can be exchanged between TK!Solver and VisiCalc, VisiFile, VisiTrend/Plot, and other programs that are DIF-compatible. Producing output in the form of a DIF file also allows the user to communicate with other programs, those written in FORTRAN or BASIC for example. The DIF file structure is published in the form of a specification, which includes sample programs in BASIC and Pascal that produce DIF files from matrices, or read DIF into matrices.

In addition, the contents of the various sheets can be printed to disk files where they may be edited with word processing programs or editors. This convention permits the incorporation of all the sheets of a TK model in an engineering report with an article and other elements, so that the entire document can be printed at one time.

The IBM PC version of TK!Solver requires 96KB of RAM, one disk drive, and either a monochrome or color display that is at least 40 characters wide. TK will use all available memory, up to 256KB; the program's performance is enhanced on a system with two disk drives and an 80-column display.

The first version of TK!Solver did not support the fixed-disk and pathnames features of DOS 2.0. Version 1.2 supports both. However, the copy-protection scheme requires that the user load the program from the floppy disk instead of from the fixed disk. Two files, TK.OVL and TK.HLP, may be copied to the fixed disk, which speeds the program's operation.

Other microcomputers must be 100-percent PC-compatible in order to run the PC version of TK!Solver. The MS-DOS version runs on computers that support that operating system and is recommended in cases where 100-percent compatibility is not certain; the program must be configured for the particular computer.

Configuration is a simple process. When the MS-DOS version of TK!Solver is loaded, the program signals that installation is necessary and presents a list of computers for which the program has been preconfigured. The user simply makes the appropriate selection and the program is ready to run. If a computer is not on the list, the user must write a terminal driver, using an example file supplied with the program. The

AW . . .
WHAT THE HECK!

ProDesign II
The Easy to Use CAD System!

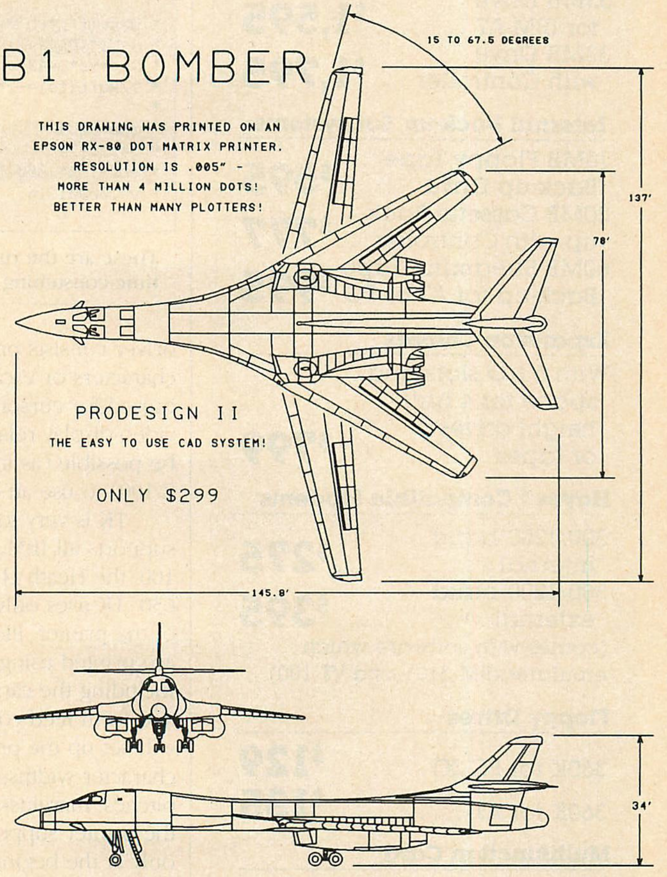
ProDesign II is one of the most advanced CAD packages available for microcomputers. We think it's absolutely the easiest to use. With competitive CAD systems priced at $1500 to $2500, we were posed with the problem of setting our price.

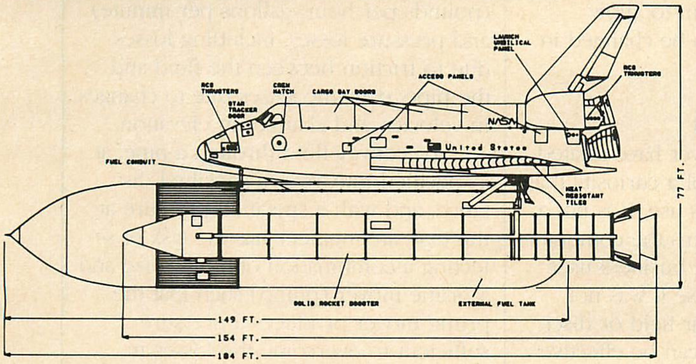
ProDesign II works a wide variety of digitizers and mouse devices. It works with nearly any plotter or printer available for the IBM PC. ProDesign II can produce plotter quality drawings on ordinary dot matrix printers - a feature found exclusively on ProDesign II. ProDesign II utilizes a virtual screen 4 times the size of the physical screen to make it practical to produce drawings on a normal resolution IBM monitor. ProDesign II is truly an outstanding CAD package for the IBM PC and compatibles. The question we had to answer was: Even though we had a better product, should we price it higher than the other CAD systems on the market?

We did market studies and calculations. We consulted with experts. We drew charts and graphs. We used the finest spreadsheet programs money could buy. When it came right down to it, we still didn't know what to sell ProDesign II for. $2995? $2495? $1995? We even considered $995.

Then, in the great American tradition, we said, "AW...WHAT THE HECK! Let's see the other guys beat this price!" ProDesign II costs $299. At that price, you can't go wrong!

AW...WHAT THE HECK!
$299.95





THE SPACE SHUTTLE DISCOVERY

Why should you get ProDesign II? Four simple reasons:

1. ProDesign II is easy to use. You won't have to spend weeks learning simple functions.
2. ProDesign II works with the hardware YOU own. ProDesign II supports most printers and plotters available for the IBM PC, as well as a wide variety of digitizers and mouse devices.
3. ProDesign II can produce plotter quality output on ordinary dot matrix printers. (The B1 Bomber above was printed on an Epson RX-80.)
4. ProDesign II is priced 70% to 80% below competitive products!

What do you need to run ProDesign II? An IBM PC or compatible with 512K RAM and graphics capability.

How do you get ProDesign II? See your local computer dealer or contact us.

ProDesign II - The Easy to Use CAD System!

American Small Business Computers
118 South Mill
Pryor, Oklahoma 74361
918/825-4844

CIRCLE NO. 103 ON READER SERVICE CARD

FIGURE 2: *Sample Rule Sheet*

```
(1r) Rule: dP=(f*L*rho*V^2)/(D*g*2)                                   85 /!

================== RULE SHEET ===========================================
S Rule
- ----
* dP=(f*L*rho*V^2)/(D*g*2)
* Q=V*(D/2)^2*pi()
* Re=V*D/v
* 1/sqrt(f)=-2*log(epsilon/(3.7*D)+2.51/(Re*sqrt(f)))

* Ds40=sch40id(D)
* Dnom=noms40(Ds40)

* dPs=(f*L*rho*Vs^2)/(Ds40*g*2)
* Q=Vs*(Ds40/2)^2*pi()
* Res=Vs*Ds40/v
* 1/sqrt(fs)=-2*log(epsilon/(3.7*Ds40)+2.51/(Res*sqrt(fs)))
*
* rho=62.4
* g=32.174
* epsilon=.00015
* v=.00001216
```

These are the rules that model the flow of arbitrary fluids in closed conduits. A time-consuming series of equations is performed automatically by TK!Solver.

driver consists primarily of control characters or escape sequences required for cursor addressing and other video-display-related tasks. It also may be possible (as in the case of a Zenith Z-100) to use an emulator program.

TK is very tolerant of printers. It supports all IBM printers, the Epson RX-100, the Heath H-25, and the Diablo 630. TK uses only the TTY-like functions of the printer. Plots, tables, and sheets are printed using only ASCII characters, including the carriage return, line feed, and form feed control characters. TK will set up the printer for various fonts, character widths, horizontal and vertical pitches, margins—almost any function the printer supports. But it will do so only at the beginning of the print command. TK does not seem to allow printer configuration to be changed in the middle of a page.

PRACTICAL MODELS

Most reviews of TK!Solver have treated this program as more of a curiosity than as a product for serious use. It is not a program that overwhelms the computer hobbyist or the average business user with its potential because it was not designed for a particular field or discipline. TK is a tool that can be effective in any situation that can use mathematical equations in a modeling process. And it speaks a language most users already know—mathematics.

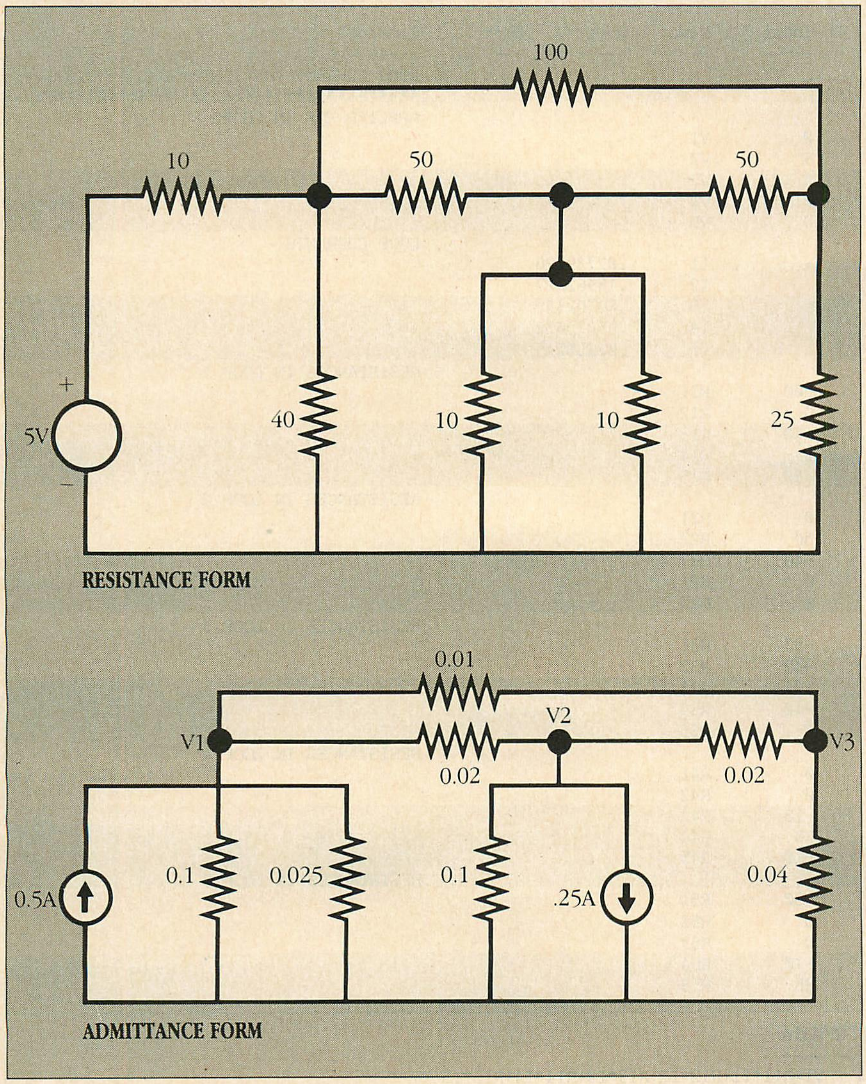
Several examples follow that demonstrate the practical use of TK!Solver modeling. Along with an explanation of each model is a description of the technique replaced by the TK model. TK!Solver *does not* presume to teach engineering or economics; it is simply a tool for the manipulation of data to form models. An engineer faced with the task of sizing piping systems and components can perform the task much faster with TK!Solver than he could using the *Cameron Hydraulic Data* book (Westaway, 1981), or a fluid mechanics book and a slide rule, or even those books and a PC running BASIC. But TK does not teach the principles of fluid mechanics.

Mechanical engineering. The first model, a pipe-sizing problem, is from *TK!Solver for Engineers* (Wright, 1984). The task is to size the piping necessary to convey fluids (gases, liquids, and supercritical fluids) and involves calculating mass or volumetric flow rates (pounds per hour, gallons per minute) and pressure losses, including losses due to friction between the fluid and the pipe, dynamic losses due to changes in velocity, and changes in elevation.

To convey fluid through a pipe, at a specified rate, over a specified distance, and with a specified pressure at the exit, an engineer has the task of selecting a combination of a pipe size and a prime mover (pump) such that the prime mover produces a pressure sufficient to overcome the losses in the pipe and still provide the required residual pressure.

The rules that model the flow of arbitrary fluids in closed conduits are shown in figure 2. These equations are published in numerous textbooks, handbooks, and catalogs. However, most engineers do not have the time to size piping systems by repeatedly solving equations. Instead, they use charts,

FIGURE 3: *Sample Circuit*



This circuit is shown graphically in resistance and admittance form. Circuits can be analyzed in terms of loop current (see figure 4) or node voltages (see figure 5).

tables, nomographs, or slide rules based on these equations; the equations themselves are generally too cumbersome to use manually.

For example, the Colebrook equation (the fourth equation in the rule sheet) cannot be factored in the variable of interest, *f* (the friction factor). Consequently, whether the engineer uses a slide rule, a programmable calculator, or a mainframe computer, the equation must be solved by an iterative procedure—just like the one TK applies automatically when it encounters this equation. A canned pipe-sizing program would also incorporate an iterative method for solving the Colebrook equation, or it will use selected values of *f* from external sources.

This particular pipe-sizing model can be used in either of two modes. In the first, the user enters a set of input values, issues the action command, and obtains a single solution, much as he would do with the *Cameron* handbook. In the second, he enters lists of values for the input variables and invokes the list processor to produce all the solutions for an entire piping system.

Electrical engineering. Electrical and electronic circuit design problems commonly require the solution of sets of simultaneous equations. Electronic circuits also are referred to as networks and their analysis as network analysis.

Circuits can be analyzed in terms of loop currents or node voltages. In either case, the model consists of a set of simultaneous equations. The circuit shown in figure 3 is modeled in loop current form in figure 4 and in node voltage form in figure 5.

Solving this problem in either way without the use of a computer would

FIGURE 4: *Circuit in Loop Current Form*

```
St Input    Name    Output    Unit     Comment
-- -----    ----    ------    ----     -------
                                       LOOP CURRENT ANALYSIS MODEL (5 LOOPS)
                                       ************************************
                                       APPLIED EMF IN LOOPS
   0        V1
   5        V2
   0        V3
   0        V4
   0        V5
                                       LOOP CURRENTS
            I1      .02739726
            I2      .16962127
            I3      .08702659
            I4      .05479452
            I5      .02256245
                                       RESISTANCES IN LOOP 1
   200      R11
   0        R12
   -50      R13
   0        R14
   -50      R15
                                       RESISTANCES IN LOOP 2
   0        R21
   50       R22
   -40      R23
   0        R24
   0        R25
                                       RESISTANCES IN LOOP 3
   -50      R31
   -40      R32
   100      R33
   -10      R34
   0        R35
                                       RESISTANCES IN LOOP 4
   0        R41
   0        R42
   -10      R43
   20       R44
   -10      R45
                                       RESISTANCES IN LOOP 5
   -50      R51
   0        R52
   0        R53
   -10      R54
   85       R55

S Rule
- ----
  V1=R11*I1+R12*I2+R13*I3+R14*I4+R15*I5
  V2=R21*I1+R22*I2+R23*I3+R24*I4+R25*I5
  V3=R31*I1+R32*I2+R33*I3+R34*I4+R35*I5
  V4=R41*I1+R42*I2+R43*I3+R44*I4+R45*I5
  V5=R51*I1+R52*I2+R53*I3+R54*I4+R55*I5
```

This set of equations is TK's loop current model for the circuit in figure 3.

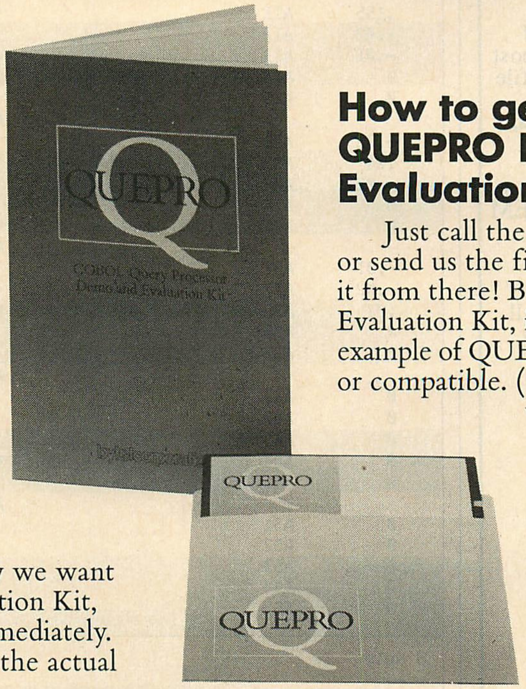
probably involve the construction of a model on paper, consisting of a coefficient matrix and a constant vector. The engineer would apply Cramer's Rule, Gaussian elimination, Gauss-Jordan elimination, or some other technique to solve the system of simultaneous equations. With a computer, a canned program could perform loop current analysis or node voltage analysis. The user also could write or possibly find some programs in a high-level language to solve the simultaneous equations.

TK!Solver requires only that the user type in the equations and give the action command. However, he must provide guess values for one or more variables because all of the equations have the same number of variables, which is a condition that requires the use of the iterative solver.

Users may apply the same program to solve any number of other electrical/electronic models, for example, to model the power factor of an electrical power system. The power factor is a measure of the balance of capacitive and inductive impedances in an electrical system. The subject is too complex for this review, but its importance is easy to make clear. For example, power companies levy special charges on customers with poor power factors. If a plant were to correct its power factor (by installing capacitor banks), the savings could, in a short time, cover the purchase of a computer and TK!Solver to avert the problem's return.

FREE FOR COBOL USERS!

A demo of our revolutionary query processor–QUEPRO.



Now, get the COBOL data you want, when you want it!

How would you like to unlock valuable COBOL data without time-wasting programming?

Or create a comprehensive report in minutes instead of days?

It's easy to do this (and a lot more), *if* you use our powerful, new query processor–QUEPRO.

But talk is cheap, and that's why we want to put a complete QUEPRO Evaluation Kit, including a demo, in your hands immediately. (We're talking about an example of the actual product–*not* a simulation!)

We want to prove that QUEPRO really *can* unlock your valuable COBOL data and put you in complete control!

How QUEPRO works.

It's easy.

Let's say you or someone in your company needs some information. Instead of placing requests with a programmer or DP Manager, holding meetings, and waiting for the product (complete with a bug or two), you can put QUEPRO to work yourself. Instantly.

That's right.

There's no programming. No command language to learn. Nothing between you and the data.

All you have to do is paint the design of your query *interactively* on the screen, choose from two formats–FORM (for screens or pre-printed forms) or LIST (for a list query or report). QUEPRO does the rest.

That means there's no need to define field formats. (Easy to remember synonyms are used to refer to data in your COBOL files.) You'll also find that complex queries involving multifile access and data transfers are a snap with QUEPRO, too!

How to get your free QUEPRO Demo and Evaluation kit.

Just call the toll-free 800 number below, or send us the filled-in coupon. We'll take it from there! But please note. This special Evaluation Kit, including a fully functional example of QUEPRO, requires an IBM-PC® or compatible. (QUEPRO, itself, runs on a wide range of operating systems including all PC-DOS and MS-DOS, Unix, Xenix, CP/-86, IMOS, and IRX.)

Whether you're a potential end-user, computer systems manager, or technical person involved with COBOL, you owe it to yourself to get in touch and take a look at QUEPRO for yourself.

There's no cost or obligation of any kind. Why not call right now?

Call toll-free 800-631-2229
or 415-527-1157.

bytelcorporation

1029 Solano Avenue, Berkeley, CA 94706, (415) 527-1157

Rush me a QUEPRO Demo and Evaluation Kit absolutely FREE! I am under no obligation of any kind. I understand that the demo runs only on an IBM-PC® or compatible, though the actual QUEPRO product runs on a variety of operating systems.

COBOL used (please check): ☐ RM/COBOL ☐ MicroSoft ☐ Microfocus ☐ Other

Name ____________________ Title __________

Company ______________________________

Address ______________________________

City ______________ State ________ Zip ________

Phone (   ) ________________________

CIRCLE NO. 204 ON READER SERVICE CARD

FIGURE 5: *Circuit in Node Voltage Form*

```
St Input    Name    Output      Unit    Comment
-- -----    ----    ------      ----    -------
                                        NODE VOLTAGE ANALYSIS MODEL (5 NODES)
                                        ************************************
                                        CURRENT SOURCES
   .5       I1
   -.25     I2
   0        I3
   0        I4
   0        I5
                                        NODE VOLTAGES (REL TO REF NODE)
            V1      3.1426269
            V2      -.7604754
            V3      .23166801
   0        V4
   0        V5
                                        ADMITTANCE MATRIX ROW 1
   .155     A11                         ** DIAGONAL ELEMENT **
   -.02     A12
   -.01     A13
   0        A14
   0        A15
                                        ADMITTANCE MATRIX ROW 2
   -.02     A21
   .24      A22                         ** DIAGONAL ELEMENT **
   -.02     A23
   0        A24
   0        A25
                                        ADMITTANCE MATRIX ROW 3
   -.01     A31
   -.02     A32
   .07      A33                         ** DIAGONAL ELEMENT **
   0        A34
   0        A35
                                        ADMITTANCE MATRIX ROW 4
   0        A41
   0        A42
   0        A43
   0        A44                         ** DIAGONAL ELEMENT **
   0        A45
                                        ADMITTANCE MATRIX ROW 5
   0        A51
   0        A52
   0        A53
   0        A54
   0        A55                         ** DIAGONAL ELEMENT **

S Rule
- ----
  I1=A11*V1+A12*V2+A13*V3+A14*V4+A15*V5
  I2=A21*V1+A22*V2+A23*V3+A24*V4+A25*V5
  I3=A31*V1+A32*V2+A33*V3+A34*V4+A35*V5
  I4=A41*V1+A42*V2+A43*V3+A44*V4+A45*V5
  I5=A51*V1+A52*V2+A53*V3+A54*V4+A55*V5
```

This TK node voltage model for the circuit in figure 3 would require the iterative solver. The user would provide guess values for one or more of the variables.

FIGURE 6: *Possible "Database" Model*

FUNCTIONS

NAME	DOMAIN	MAPPING	RANGE	COMMENT
Fstreet	Name	Table	Street	
Fcity	Name	Table	City	
Fstate	Name	Table	State	
Fzip	Name	Table	Zip	

RULES

Addr1 = Fstreet (Name)
Addr2 = Fcity (Name)
Addr3 = Fstate (Name)
Addr4 = Fzip (Name)

TK can include user functions that serve as look-up tables. By constructing a series of functions, such as the above, a user could produce a model that is perhaps not a substitute for a real database manager, but may be a great improvement over a box of three-by-five cards. TK's strength is in a complete and accurate application.

Financial and economic. TK is equally at home with financial and economic problems to the extent that they can be modeled mathematically. A model could be created that would contain a set of discount factors used to solve cash flow problems known as equivalence problems. This model would evaluate the worth of various alternative cash flows.

TK is useful to many disciplines—its strength is in a complete and accurate application. Remember that a TK model can include user functions that serve as look-up tables. By constructing a series of functions, a user could produce a model that is perhaps not a substitute for a real database management program, but indeed may be a great improvement over a box of three-by-five cards (see figure 6).

TK can expand determinants, perform numerical integration and differentiation, find maxima and minima of functions, interpolate between discrete points of a function, fit curves to data sets, solve the characteristic equation of a differential equation, describe functions as Fourier series, and perform Laplace transformations.

A FEW LIMITATIONS

TK!Solver has some computational limitations. It will not perform symbolic differentiation or integration. In fact, it does not perform symbolic manipulation of any sort. This means the user cannot enter an arbitrary function and instruct TK to find the integral or derivative of the function. Nor can he ask TK to solve an equation in terms of symbols. The programs that do perform symbolic manipulation on a microcomputer are more difficult to use than TK!Solver and would require considerable programming to do the many other tasks that TK performs easily.

In another area, TK often is forced to use the iterative solver because of the way in which a model is constructed, even though an alternative construction of the same set of rules might not force iteration. However, it may not be fair to call this a limitation because the manual explains the conditions required to use the direct solver and they are not stringent. It is preferable to construct a model so that it can be solved with the direct solver.

TK!Solver's most serious limitation is its lack of explicit programming constructs: IF . . . THEN, REPEAT . . . UNTIL, DO . . . WHILE, GOTO. In TK's favor is that its overall use does not demand a knowledge of a certain programming language, nevertheless, some circumstances are handled more easily

in terms of inequalities and procedures than in terms of equations.

However, TK can be "programmed" to behave as if it does include CASE, IF . . . THEN, and REPEAT . . . UNTIL statements. A model can be constructed that branches based on logical conditions, using the program's built-in STEP function or by including logical variables in the rules. The model shown in figure 7 sizes ducts, for either laminar or turbulent flow, with the use of a logical variable.

Not all models that require iteration will converge to a solution in a well-behaved manner. In some cases, failure to converge is a consequence of the manner in which TK forms successive guess values; in other cases, it is a consequence of the model itself or of the initial guess value. If TK does not converge properly, it will exhibit one of three behaviors: slow convergence, divergence, or oscillation.

Slow convergence is less of a problem than the other two. If TK fails to reach a solution in the number of iterations it is set to perform (default value is 10), the solution process can be resumed by pressing the action key again. TK resumes convergence, but with the last guess of the previous session as the initial guess value. If a particular model is prone to slow convergence, the maximum iteration count field on the global sheet can be increased.

Divergence is the opposite of convergence—that is, successive guesses that are produced are further from the solution instead of closer to it.

In order to tell if a model is oscillating, set the intermediate redisplay on field of the global sheet set to yes. The guess variable will alternate between two values or will alternate between positive and negative values of increasing absolute magnitude.

Some reviewers have listed slowness as a TK!Solver limitation. This may be an unfair assessment. It takes longer, of course, to solve larger models than smaller models, but this is parallel to electronic spreadsheets that may take longer to recalculate large worksheets or word processors requiring extra time to reformat large documents. The trade-off is in the *number* of tasks TK can perform. Some canned engineering programs, written in assembly language or FORTRAN, are very fast but solve only a single type of problem.

Performing benchmark tests to measure speed is not appropriate in this case. For example, a comparison between TK running a pipe-sizing model and a canned program working

FIGURE 7: *Duct-sizing Model*

St	Input	Name	Output	Unit	Comment
					static regain duct sizing model
L		Pv	.08	lbf/ft^2	velocity pressure
L		V	499.8	ft/min	velocity of fluid in duct
L	500	Q		ft^3/min	air flow rate
L		A	1.0004002	ft^2	cross-sectional area of duct
L		dPf	.26186464	lbf/ft^2	friction loss
L		f	.03694272		friction factor, dimensionless
L	100	L		ft	length of duct
L		D	1.1286049	ft	duct diameter
		rho	.075	lbm/ft^3	air density
L		Re	5767.1274		Reynolds number
		mu	.0000038	lbf*s/ft^	fluid dynamic viscosity
		lam	0		1 if Re <= 2000, 0 otherwise
		trans	1		1 if Re > 2000, 0 otherwise
	.001	epsilon		ft	absolute roughness of duct wall
L		W	16.973958	in	duct width
L		H	8.4869789	in	duct height
		Tlength	200	ft	total equivalent length of duct path
		TPf	.11530388	in_wc	total friction loss in path
	2	ar			aspect ratio of a rectangular duct
L		IDsq	12.002401	in	inside dimension of square duct
L		prev			
	.09617234	Pmin		in_wc	minimum static pressure
L		Dl	162.51911	in	duct diameter in inches
		first	0		
L		Qcurren	8.3333333	ft^3/min	
		Ps	3	lbf/ft^2	
L		Pt	2.6795	lbf/ft^2	
		gc	32.174	lbm*ft/lb	conversion factor
L		Lt	80	ft	running length

```
S Rule
- ----
* prev=element()-1
* first=step(1,element())
* first*  V=8.33  *first
* Lt=L+element('Lt,prev,0)
* gc=32.174
* Ps=3
* first*  Pv=.08  *first
* (1-first)*  Pv=element('Pv,prev,0)+element('dPf,prev,0)  *(1-first)
* Pt=Pv+Ps
* (1-first)*    V=sqrt(Pv*2*gc/rho)    *(1-first)
* A=Q/V
* D=2*sqrt(A/pi())
* rho=.075                        "standard air density
* mu=3.8E-6                       "fluid dynamic viscosity at 75'F
* Re=rho*D*V/(mu*gc)  "D is in ft, V in ft/sec, mu approx 3.8x10^-6
* lam=step(2000,Re)
* trans=step(Re,2001)
* lam* f=(64/Re)  *lam
* trans*  1/sqrt(f)=-2*log((epsilon/(3.7*D))+2.51/(Re*sqrt(f)))  *trans
* dPf=f*(L/D)*Pv
* Dl=D
* IDsq=sqrt(A)
* Tlength=sum('L)                 "Total equivalent length of duct path
* TPf=sum('dPf)                   "Total friction loss in path
* H=sqrt(A/ar)
* W=ar*H
```

From	To	Multiply By	Add Offset
ft	in	12	
ft^2	in^2	144	
in_wc	lbf/ft^2	5.199	
min	s	60	
ft/s	ft/min	60	
ft^3/s	ft^3/min	60	

TK!Solver's most serious limitation is its lack of programming constructs, however, a TK model can be created that branches based on logical conditions, using the built-in STEP function or by including logical variables in the rules. This model sizes ducts for either laminar or turbulent flow, with the use of a logical variable.

on the same problem is irrelevant because there is no way of knowing if the canned program is performing the same functions as the TK model.

An evaluation of TK's speed must consider the time saved in the total modeling task. If a user chooses to compare TK to a program compiled in

FORTRAN, he should add the programming time to the runtime of the program. The rules (or equations) must be developed in either case, and debugging is often required with a TK model. Another consideration is that it is not necessary to know FORTRAN or any other language to use TK. If a user knows the mathematics required to construct a model on paper, he can construct it in TK.

The PC version of TK is limited in its use of graphics. TK's plots are quite crude when compared to graphs produced by other spreadsheet programs. The user has some control over the quality of the plot produced on the printer, but no control over what is produced on the screen. (This may be understandable considering TK's background—as a program that ran on a minicomputer with a TTY-like terminal.) TK is strictly character-oriented.

THEORETICAL CONSIDERATIONS

Understanding how TK!Solver works is not necessary to use it effectively. Aside from the manual, little has been written about the product. TK grew out of an equation-solving algorithm called consecutive substitution procedure (CSP) and a larger program called the question answering system on mathematical models (QAS), based on CSP.

Backsolving is a term that will be familiar to any TK!Solver enthusiasts who have read more than just the product's manual. It refers to solving for unknowns on the right side of the equal sign as well as for those on the left side. An equation is considered "backsolvable" if there is a legal sequence of TK arithmetic operations and functions that allow it to isolate the unknown on one side of the equal sign.

TK does not solve a model in quite the way that might be expected. An equation can be rearranged symbolically until the unknown is on the left side of the equal sign, then known values are substituted for the other variables. TK cannot perform the symbolic rearrangement; it takes the equation *as is*, substitutes known values, and simplifies the equation arithmetically. When a rule is simplified to the point that a variable name appears on one side of the equal sign, and a numeric or symbolic value appears on the other, that variable becomes known and is available for substitution into other rules. When the last variable becomes known, the model has been solved.

Each rule (or equation) must meet two conditions to be backsolvable with respect to a variable: (1) the variable of interest must appear only once in the equation, and (2) the variable of inter-

BETTER STILL

TK!Solver is not without its limitations, but Software Arts appears to be responding to the comments of TK users, reviewers, and other critics. Sources at the company confirm that a new version of TK!Solver, informally called TK2, will be introduced in the near future. A few of the more significant enhancements are listed below:

- Instead of the prompt line of single characters, TK2 will display a pull-down menu of commands when the slash character is typed. The command can be completed by typing the first letter of the command, as with TK, or by moving the cursor to the desired command via the arrow keys, and hitting Return.
- Rules can be temporarily removed from a model by entering a C in the status field on the rule sheet. (TK requires that the rule be commented out by entering quotation marks with the Edit command.)
- TK2 will use high-resolution graphics for plots if a computer is suitably equipped. If it is not, TK2 will display character plots, as does TK.
- TK2 will handle complex numbers in addition to real numbers and symbolic values.
- TK2 will include a numeric format sheet to control the display format of numeric values.
- TK2's list-filling function will provide the option to fill a list with a geometric progression in addition to arithmetic progression.
- TK2 will plot portions of lists and provide logarithmic axes as well as linear axes.
- TK2 will display tables with row and column separators, and will display portions of lists.
- TK2 will allow conditional rules, such as IF . . . THEN.
- User functions under TK2 can be defined as list functions or as procedural functions.
- TK2 can be set to run in color.
- TK2 will use more memory.
- TK2 can coexist with other programs, such as Spotlight.

—*VEW*

est cannot be the argument of a function without a unique inverse.

The pipe-sizing model illustrates the first condition in the Colebrook equation. The friction factor variable, *f*, appears twice and the equation cannot be rearranged to eliminate the condition. The second condition can occur when equations contain functions such as INT(x), ABS(x), or SGN(x).

The iterative solver is the part of TK that makes the program practical for real-world engineering problems; it is also responsible for giving the impression of slowness. Again, the speeds of the two solvers are only relative. A TK model that requires iteration represents a problem that would be a time-consuming proposition for any program. (Try, for example, to duplicate the performance of the pipe-sizing model with a spreadsheet program.)

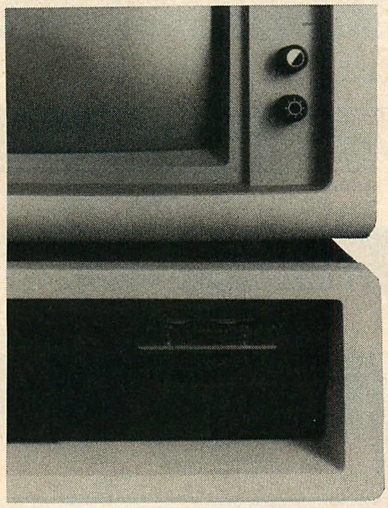
Considering that TK models can be stored for repeated use and that most users will take advantage of this facility, the initial design of a model should reflect some thought to minimize the need for iteration. A model should be built to take advantage of the direct solver whenever possible. Often this is accomplished by including redundant equations, which are combinations of other equations in the model. They do not add any information; they merely create conditions that are favorable to the direct solver.

TK's authors have developed techniques to aid in TK model design. Two of these techniques are presented in *The TK!Solver Book* (Konopasek and Sundaresan, 1984); they are variable-equation coincidence (VEC) matrices and R-graphs. Both techniques are useful in optimizing models.

In addition to its being an equation-solving program that uses the language of mathematics, TK represents a concept called a *constraint language* or a *declarative language*.

Most programming languages are procedural. To solve a problem in FORTRAN, the user begins with a set of constraints, or conditions, such as a set of equations (which could become the rules of a TK model). But, these constraints are not acceptable to the FORTRAN compiler. The user must create an algorithm and translate equations into legal FORTRAN statements. Both the algorithm and the final set of FORTRAN statements represent a set of steps the computer must take to provide a solution to the problem—this is the essence of a program.

With a constraint language, the solution algorithm is left to others. In-



stead, the user concentrates on describing the problem in terms of constraints: for example, if the Reynolds number is less than 2,000, the flow is laminar, or, in the language of TK:

```
Re = rho*D*V/(mu*12*32.174)
lam = step(2000, Re)
```

With the exception of the step function, these two statements should be immediately understandable to the student of fluid mechanics because TK's constraint language is basically the same mathematics studied by all engineers.

TK!Solver represents a significant advance in its use of constraint and declarative languages. In the real world of engineering, it accepts equations as engineers have learned them, not as they have been interpreted by computer programmers. While integrated packages may produce better graphs and database managers do a better job of accounting, there are still some engineering tasks that are better solved by algorithmic programming. TK!Solver may finally replace some of these tools. Even now, it a useful addition to any serious microcomputer library.

REFERENCES

Konopasek, Milos and Jayaraman Sundaresan. *The TK!Solver Book*. Berkeley: McGraw-Hill, 1984.

Westaway, V. R. and A. W. Loomis, editors. *Cameron Hydraulic Data*. Woodcliff Lake: Compressed Air Magazine Company (division of Ingersoll-Rand Company), 1981.

Wright, Victor E. *TK!Solver for Engineers*. Reston: Reston Publishing Company, 1984.

Between the time that PC Tech Journal *accepted this article and the time it was published, Lotus bought Software Arts. Lotus has announced that it will continue to support and market TK!Solver, but it is unclear which channels of distribution it will use. Rumors suggest that Lotus will distribute the product through OEMs exclusively, which could mean an IBM-label version of TK!Solver in the future. This rumor also implies that the product will be removed from the retail shelf.*

We regret that we are unable to provide more detailed information at this time. Watch the Tech Releases department for news of developments.

—WF

Victor E. Wright is the manager of process engineering at Luckett & Farley in Louisville, Kentucky. He has a degree in industrial management from the University of Cincinnati.

AN EDITOR IN THE ALLEY

The editor of *PC Tech Journal* challenged TK!Solver to solve the following problem: The editor has two ladders, one 30 feet long, the other 40 feet long. He wishes to arrange them in an alley that is bounded by skyscrapers such that they cross 10 feet above the ground. What should the width of the alley be?

The listing at right is a BASIC program, written by the editor, that solves this problem iteratively. The program requires a guess value for the alley width, and checks to see whether the guess is legal; negative entries are not allowed, nor are guesses over 30 feet. The program converges nicely on the correct answer, given any legal guess; it will not accept illegal guesses.

This problem should have been solved easily by TK!Solver's iterative method, but it was not. Seven models were produced that ran with varying degrees of success; all converged on a solution, given a guess within a certain range, but none could handle the range of guesses the editor's program could handle. Observation of the intermediate values displayed by TK indicate that the problem lay in the manner in which TK formulates its guess values.

This exercise would seem to cast doubt on TK!Solver. However, the situation described above illustrates only one method of iterative problem solving available to the TK user, and the editor had complete control over the method of iteration he used. In addition, TK had to use its standard method, or did it?

The TK model shown in the figure with this sidebar demonstrates

FIGURE 1: *TK! Solver Solution Model*

```
(1r) Rule: pass1 = step(1,element())                                   461/!

=================== RULE SHEET ==============================================
S Rule
- ----
* pass1 = step(1,element())
* remainder = step(element(),2)
* WG = pass1 * guess + remainder * nextguess
* Y1 = sqrt(900 - WG^2)
* Y2 = sqrt(1600 - WG^2)
* slope1 = Y1 / WG
* slope2 = Y2 / WG
* X1 = 10 / slope1
* X2 = 10 / slope2
* W = X1 + X2
* error = WG - W
* nextguess = element('WG,element()-1,0) + element('error,element()-1,1)/2

(11c) Comment: Guess for W                                             461/!

=================== VARIABLE SHEET ==========================================
St Input      Name    Output     Unit       Comment
-- -----      ----    -------    ----       -------
                                            LADDERS IN THE ALLEY MODEL
                                            **************************
L             slope1                        Slope of ladder 1
L             Y1                            Height of ladder 1, upper end
   28         guess                         Initial guess
L             slope2                        Slope of ladder 2
L             Y2                            Height of ladder 2, upper end
L             X1                            Horizontal distance to intersection
L             X2                            Horizontal distance to intersection
L             W                             Width of alley
L             nextgue                       Guess for W
L  1          count                         Counter
L             WG                            Guess for W
L             pass1                         Logical flag
L             remaind                       Logical flag
L             pass2                         Logical flag
L             error                         error term

(1n) Name: count                                                       461/!

=================== LIST SHEET ==============================================
Name       Elements  Storage Unit  Comment
----       --------  ------------  -------
count      30
nextguess  30
W          30
X2         30
X1         30
Y2         30
slope2     30
guess
Y1         30
slope1     30
WG         30
pass1      30
remainder  30
pass2
error      30
```

This model uses TK's direct solver under the supervision of the list processor.

that the TK user is not limited. Instead of using TK's iterative solver, this model uses the direct solver under the supervision of the list processor. The variables of interest are associated with lists, and one additional list, called count, does not appear as a variable in a rule. Count is just that, a counter. It is the only list set up as an input list, and it will cause the list processor to invoke the direct solver once for each element, even though the values of the list are not used in calculations.

The model includes rules to determine whether the model is making the first pass, or any pass following the first one. On the first pass, the initial guess is used in the calculations; on subsequent passes, the results of the immediately preceding pass are used. The model makes an error calculation and forms the next guess with the aid of the error value.

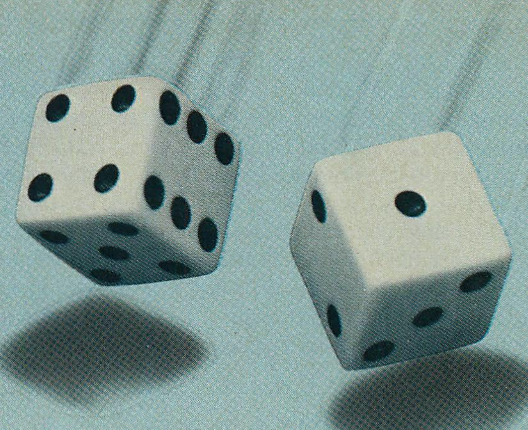
The model is constructed so that the direct solver can solve each rule using values that were obtained in preceding rules, in a single pass. In this case, iterative solver is not required and the model converges nicely for initial guess values from 1 to 29 plus.

—*VEW*

LISTING: ALLEY.BAS

```
100 REM Program to find alley width
110 REM (to within about 1/80th of an inch (1/1000 of a foot))
120 REM Will Fastie
130 KEY OFF: CLS
140 LADDER(1) = 40
150 LADDER(2) = 30
160 CROSS.Y = 10
170 INPUT "Enter first guess:  ",GUESS
180 IF GUESS >= LADDER(2) OR GUESS >= LADDER(1)
        THEN PRINT "    The alley cannot be that wide.": GOTO 170
190 IF GUESS <= 0
        THEN PRINT "    That alley is too narrow!": GOTO 170
200 GUESS.DELTA = 5: LAST.GUESS = GUESS + GUESS.DELTA
210 REM Calculate wall heights and slopes for this guess
220 FOR L = 1 TO 2
230   HEIGHT(L) = SQR(LADDER(L)^2 - GUESS^2)
240   SLOPE(L) = HEIGHT(L) / GUESS
250 NEXT L
260 REM Make ladder 2 slope negative (it goes the other way)
270 SLOPE(2) = -SLOPE(2)
280 REM Calculate 10 foot high points for each ladder
290 FOR L = 1 TO 2
300   CROSS.X(L) = CROSS.Y / SLOPE(L)
310 NEXT L
320 REM Adjust ladder 2 cross point into first quadrant
330 CROSS.X(2) = CROSS.X(2) + GUESS
340 REM Calculate delta, determine next guess
350 DELTA = CROSS.X(2) - CROSS.X(1)
360 F$ = "##.###    ##.### - ##.### = ###.####  --> "
370 PRINT USING F$; GUESS,CROSS.X(2),CROSS.X(1),DELTA;
380 IF ABS(DELTA) < .001 THEN PRINT "Found!": END
390 IF DELTA > 0 THEN PRINT "too narrow": GOTO 410
400 IF DELTA < 0 THEN PRINT "too wide":   GOTO 460
410 REM Alley was too narrow, make wider
420 IF LAST.GUESS > GUESS THEN GUESS.DELTA = GUESS.DELTA / 2
430 LAST.GUESS = GUESS
440 GUESS = GUESS + GUESS.DELTA
450 GOTO 210
460 REM Alley was too wide, make narrower
470 IF LAST.GUESS < GUESS THEN GUESS.DELTA = GUESS.DELTA / 2
480 LAST.GUESS = GUESS
490 GUESS = GUESS - GUESS.DELTA
500 GOTO 210
```

The stakes are high and you can't afford to gamble with pc communications.



ANNOUNCING DATA DECISIONS' PC COMMUNICATIONS

Micro Communications is one of the most technically sophisticated DP environments there is. It's also the one that affects the most people—technical and non-technical alike.

But mastering the technology and interconnecting the people is your company's key to success with pc's... If you're responsible for micros at your organization, it's also the key to success at your job.

PC COMMUNICATIONS is the *first* and *only* information service that will take the gamble out of your micro communications decisions and give you a handle on this complex new world.

Hands-on Test Results

It's a unique new One-Stop Information Source that TESTS and EVALUATES all the hardware and software products you need to link your pc's to corporate mainframes, minis, data services and other pc's.

Looking for a micro-to-mainframe link? We include everything from terminal emulation... to file transfer and data conversion... to software products that connect pc's directly to mainframe databases.

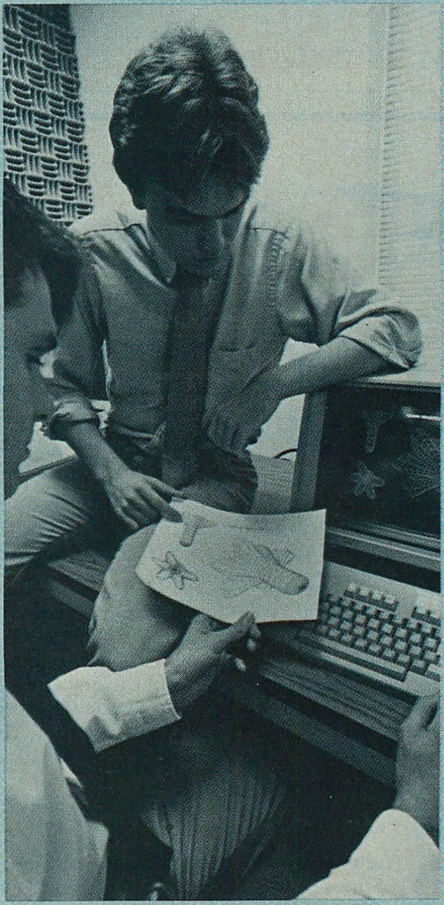
Looking at LANs? We test them with single and multiple users... with different kinds of users... with different types of software.

Want to connect your pc's to online databases—or to each other? Live hands-on test results for all the products you need to know about are here.

Your One-stop Look-up

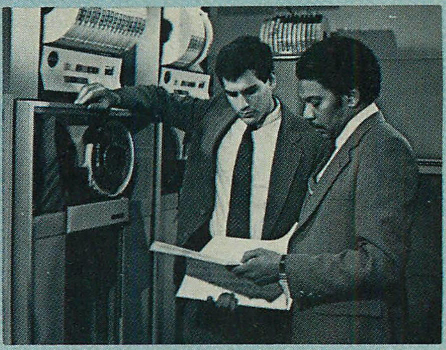
Whatever your needs—from dial-up modems to protocol converters to wideband hardwired networks—we cover them all... we test them all... and give you independent expert evaluations you can trust.

And giving you product information is just the beginning!



Our testing is done by experts. Data Decisions employs the most capable staff of professional micro communications experts in the information service industry—specialists with backgrounds in programming, consulting and corporate micro use who provide you with the highest level of expertise.

Our tests reflect your world. Our test methodologies are developed by our experts in consultation with prominent industry leaders, corporate DP managers and end users. This means that each test measures the right factors and reflects the conditions you deal with in the real world.



We do hands-on testing. Data Decisions analysts never use canned demos. We use our own data and we test standalone and LAN products in our own labs. Our analysts look at complex interconnect environments all over the country to put products through their paces while they're up and running at outside mainframe sites.

Analysis of Emerging Trends

The same experts who evaluate these products also spot emerging trends. They tell you where the new micro communications technologies are going and how they work... They also tell you what will matter in the long run—and what won't. And because you'll know what your options are at the earliest possible date, you won't have to worry that decisions made today will shut you off from future products and technologies.

What You Get

PC COMMUNICATIONS comes to you in 3 loose-leaf volumes which are updated every month to ensure their currentness and completeness.

In them you'll find the results of hundreds of hands-on tests *done under consistent, controlled laboratory conditions* by our specially trained team of micro communications experts—experts who tell you what each product's strengths and limitations are... how competing products compare... and what you can expect when you interconnect different products and turn them on.

You'll also get:

- Easy-to-read rating charts that show how suitable major hardware and software products are for novice, experienced nontechnical, and technical users.
- Action-oriented technology reports that fill you in on industry trends.
- Concise product profiles that show you what's available—in both the U.S. and Canada.
- Reliable user ratings that let you see how products measure up in actual use.
- Up-to-the-minute monthly newsletters and report updates... Complete revisions of all product reports at least once a year.
- A comprehensive vendor directory that also includes Canadian distributors.

> **"Your telephone consultants are just outstanding—a really great group. They provide a timely, personal response to virtually any type of inquiry."**
>
> **Stan Leu, Director of Marketing**
> **DS Agency, Inc., Lubbock, TX**

> "Your people do a wonderful job on the hands-on evaluations. I particularly appreciate the way you look at a product from a corporate standpoint, not just from the technical point of view."
>
> Gerald Nadeau, Systems Analyst
> Analog Devices, Burlington, MA

The books are just the beginning.

Along with all these benefits, you also get *unlimited free telephone consulting* by experienced, informed experts, primed and waiting to hear from you—and ready to track down exactly the information you need. In fact *many clients tell us the peace of mind this FREE service gives them is worth the subscription price all by itself.*

> "We act as a data resource information center for the rest of the company so we try to stay abreast of just about everything. We use Data Decisions services to answer specific questions on particular products and market sectors and find them a valuable research tool."
>
> Stephen Lofgren, Group Manager
> NEC Systems Laboratory,
> Boxborough, MA

Special No-Purchase-Obligation Trial Review Offer

A 1-year subscription—all 3 volumes complete with updates, newsletters and free unlimited phone consulting—is only $800 plus $25 shipping and handling... far less than you'd have to pay an outside consultant for even one of our in-depth evaluations.

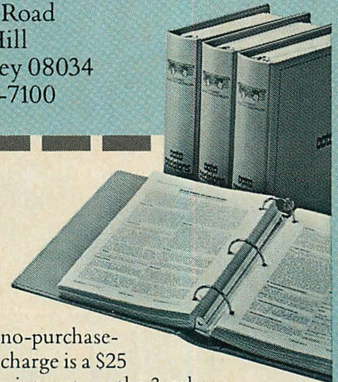
Data Decisions is so confident of your response to PC COMMUNICATIONS that we're offering to let you examine all 3 volumes *at your leisure* in your own office *with absolutely no purchase obligation.* To take advantage of this outstanding offer, simply agree to pay the shipping and handling fee. If, for any reason, you decide that PC COMMUNICATIONS isn't for you, just return the 3 volumes.

So order your trial review today. You'll also be able to see for yourself how PC COMMUNICATIONS can *save you hundreds of hours of testing and thousands of dollars in costly purchasing mistakes.*

ACT NOW!

Complete this order form and mail today!

Data Decisions
PC COMMUNICATIONS
20 Brace Road
Cherry Hill
New Jersey 08034
(609) 429-7100



Data Decisions
PC COMMUNICATIONS
20 Brace Road • Cherry Hill, NJ 08034

☐ C6 **YES!** Please send me Data Decisions PC COMMUNICATIONS for a no-purchase-obligation trial review. I understand that the only charge is a $25 shipping/handling fee and that my only obligation is to return the 3 volumes if I decide not to subscribe.

☐ Y6 Although your offer sounds great, I'm still not convinced. Please send me additional information.

Signature____________________

Name______________ Title__________

Company____________________

Address____________________

City________ State________ Zip________

Phone (_____) ________________

A202


It's easy to make a copy.
It's quick.
It's illegal.
It's wrong.
It's hard to believe.
People who wouldn't think of shoplifting a software product on their lunch hour don't think twice about going back to the office and making several illegal copies of the same software.
Making unauthorized copies of software is a violation of U.S. Copyright Law. Yet, the problem has reached epidemic proportions because many people are unaware, or simply choose to ignore the law. The software industry is urging decision-makers and software users to take steps to stop software piracy in their organizations. In the meantime, the industry has been forced to prosecute willful copyright violators.
There are legal, moral and economic imperatives forbidding theft of copyrighted software.
There is a free pamphlet on the subject. Call or write for a copy. A copy. A copy. A copy for everyone you know. Please ask for Priscilla.
ADAPSO
ADAPSO
1300 North Seventeenth Street
Arlington, Virginia 22209
(703) 522-5055

Scientific Graphics with the EGA

These primitives provide basic graphics capabilities using the EGA's 640-by-350 resolution.

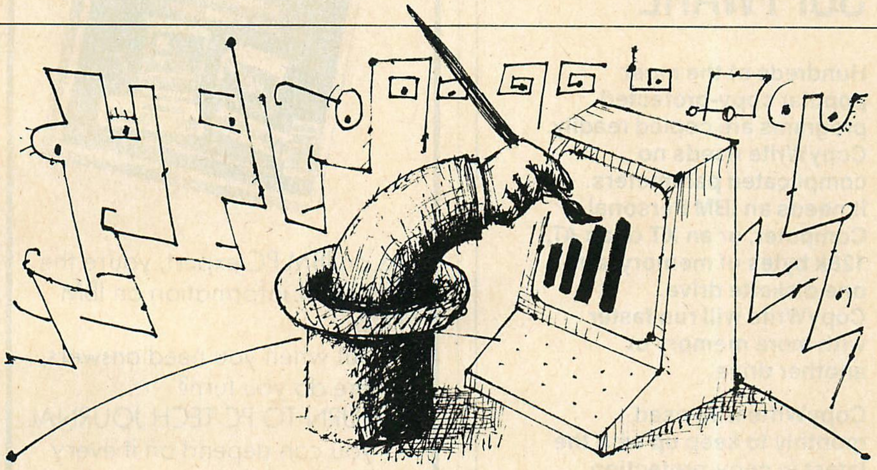


ILLUSTRATION • MACIEK ALBRECHT

A big surprise was in store for persons who ordered IBM Enhanced Graphics Adapters (EGAs) for use with standard monochrome displays. The IBM product announcement seemed to suggest that a high degree of compatibility existed between the old software and the new hardware. Being able to display scientific data in 640-by-350 resolution without having to sacrifice the high-quality text display on the monochrome monitor sounded attractive. It did not take long, however, to discover that none of the graphics software written for the old color graphics adapter would run on the monochrome monitor with the EGA.

IBM representatives have not been particularly forthcoming with technical documentation on the EGA. Other applications problems surfaced as well: Borland International has been unable to help users get Turbo Pascal graphics running on the hardware; neither can help be found among CompuServe users for doing graphics with Turbo on the EGA with the monochrome display. When the EGA is used with the standard color monitor, the usual graphics commands in Turbo work properly; but the monochrome display can operate only in the new 640-by-350 mode, for which no graphics software exists.

The article on the EGA published in the April 1985 issue of *PC Tech Journal* ("Graphic Enhancement," Thomas V. Hoffmann, p. 58) is helpful in the development of the following simple primitives, which are adequate for doing scientific graphics on the EGA with the monochrome display or the new IBM Enhanced Color Display (which also supports the 640-by-350 mode).

Once the memory locations for the video buffer in the EGA were found, building efficient routines in Turbo Pascal was straightforward. The routines in listing 1 were written with speed in mind. (Listing 2 demonstrates the primitives in listing 1.) Although they are not in assembly language, they write directly to the video memory; use only integer calculations to avoid the speed penalty of real-number arithmetic; and use the Turbo operations of shift right and left whenever possible. The actual illuminating of pixels is almost twice as fast as the standard Turbo Pascal 3.00B **plot** procedure, but the **draw** and **circle** routines are about 10 percent and 50 percent slower, respectively, than their Turbo Pascal counterparts.

The procedures are described here as they would be called in normal use. Procedure **init_graphics** is called to initialize the EGA for 640-by-350 graphics mode. The routine checks the EGA's BIOS work area for the EGA's switch settings and the monochrome-display-installed bit. It then sets the EGA graphics mode most appropriate for those settings. An important note: although **init_graphics** will set the *mode* correctly for 320-by-200, 40-column color mode (if the EGA's DIP switches have been set to select that mode), the drawing logic will not work in that mode.

Arrays containing precalculated video buffer addresses are set up within **init_graphics** to speed the addressing of individual pixels by avoiding repetitive multiplications. (This use of arrays and direct video buffer access is probably what accounts for the speed of this **plot** procedure, which is faster than the standard Turbo version.)

Plot lights up an individual pixel located with the usual *X*, *Y* coordinates. The first byte of memory defines the first eight pixels, the next byte the following eight, and so on sequentially. Because all the new EGA modes have the same starting address and sequential memory, they all can use **plot** just as it is. Notice that the byte within which the individual bit is located must be logically ORed with the present contents of memory for that byte.

Scientific graphing can be done using just **init_graphics** and **plot**. The other procedures are embellishments, although some might be considered essential. The **draw** and **circle** algorithms are adapted from Don Rollins's *IBM PC 8088 Macro Assembler Programming* (Macmillan, 1985).

Switch is a simple procedure to exchange the value of two integer variables. Procedure **draw** calls **switch** in making a straight line between any two points (*X*1, *Y*1) and (*X*2, *Y*2). A brute-force method for connecting all the points between the pair of points is merely to pick the longest axis as the independent variable and then calculate the slope. Each time the variable in the

long axis is incremented by one, the other variable is incremented by the slope until the end of the line is reached. Because the slope is a real number, this method involves costly real-number arithmetic.

Depending upon the slope of the line, the short axis is not always incremented each time the long one is. For example, if the X-axis is the longest and the slope is nearly zero (close to horizontal), the Y-axis is rarely incremented. The real-number arithmetic and the frequency with which the short axis is incremented are handled by introducing the **cycle** variable, which is the slope multiplied by half of the length of the long axis. This makes **cycle** an integer and eliminates any real-number arithmetic during the line draw.

Real numbers or not, eight separate cases must be considered in drawing a line between two end points. These consist of the following four possibilities, multiplied by the two variations depending upon whether the X-axis is longer than the Y-axis:

$(X1 > X2, Y1 > Y2)$
$(X1 < X2, Y1 < Y2)$
$(X1 > X2, Y1 < Y2)$
$(X1 < X2, Y1 > Y2)$

The two variations are handled by switching the meaning of X and Y if the Y-axis is the longer one. The essential ingredients of the **plot** procedure are included within **draw**; this increases the speed of the procedure by about one-third by eliminating large numbers of procedure calls to **plot**.

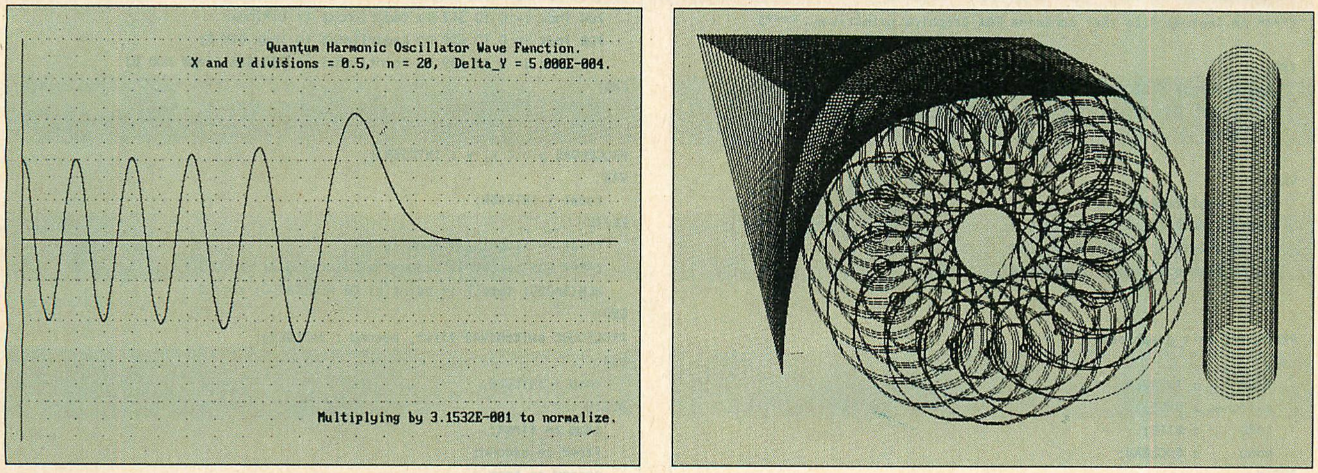
Procedure **circle** also calls **switch**, and function **isqrt** (described below) finds integer square roots through a method that avoids all real-number arithmetic. The brute-force method for drawing circles defines the angle theta as being between the radius vector and the X-axis and increasing in the counterclockwise direction. Then the X-coordinate of the circle equals the *radius* times *cos*(theta), and the Y-coordinate equals the *radius* times *sin*(theta). Evaluation of transcendental functions is slow, although the 8087 helps. The following equation is better for drawing a circle:

$$Y = round(sqrt(sqr(radius) - sqr(X)))$$

This is the fastest method to use if an 8087 is installed; without an 8087, using the function **isqrt** is almost as fast.

The idea for **circle** is to start with $X = radius$, find Y, decrease X by 1, find a new Y, and so on. Full use is made of the circle's symmetry to plot

FIGURE 1: *Examples of output*



These examples were printed on the IBM 5182 Color Printer using screen_dump, which preserves screen pixel proportions.

eight points for each evaluation of the equation for a circle. The interior WHILE loop takes care of the inevitable straight-line segments that occur in the four cardinal directions. Because a mathematical circle is displayed as an ellipse on the screen, an aspect ratio has to be used. To avoid real-number arithmetic, this is done by multiplying the *X*-coordinates by 120 and then doing an integer divide by 100.

Isqrt finds integer square roots by subtracting odd integers from the argument until it is driven negative. With a little modification to take care of overshoots, the root is then one-half of the most recent odd integer.

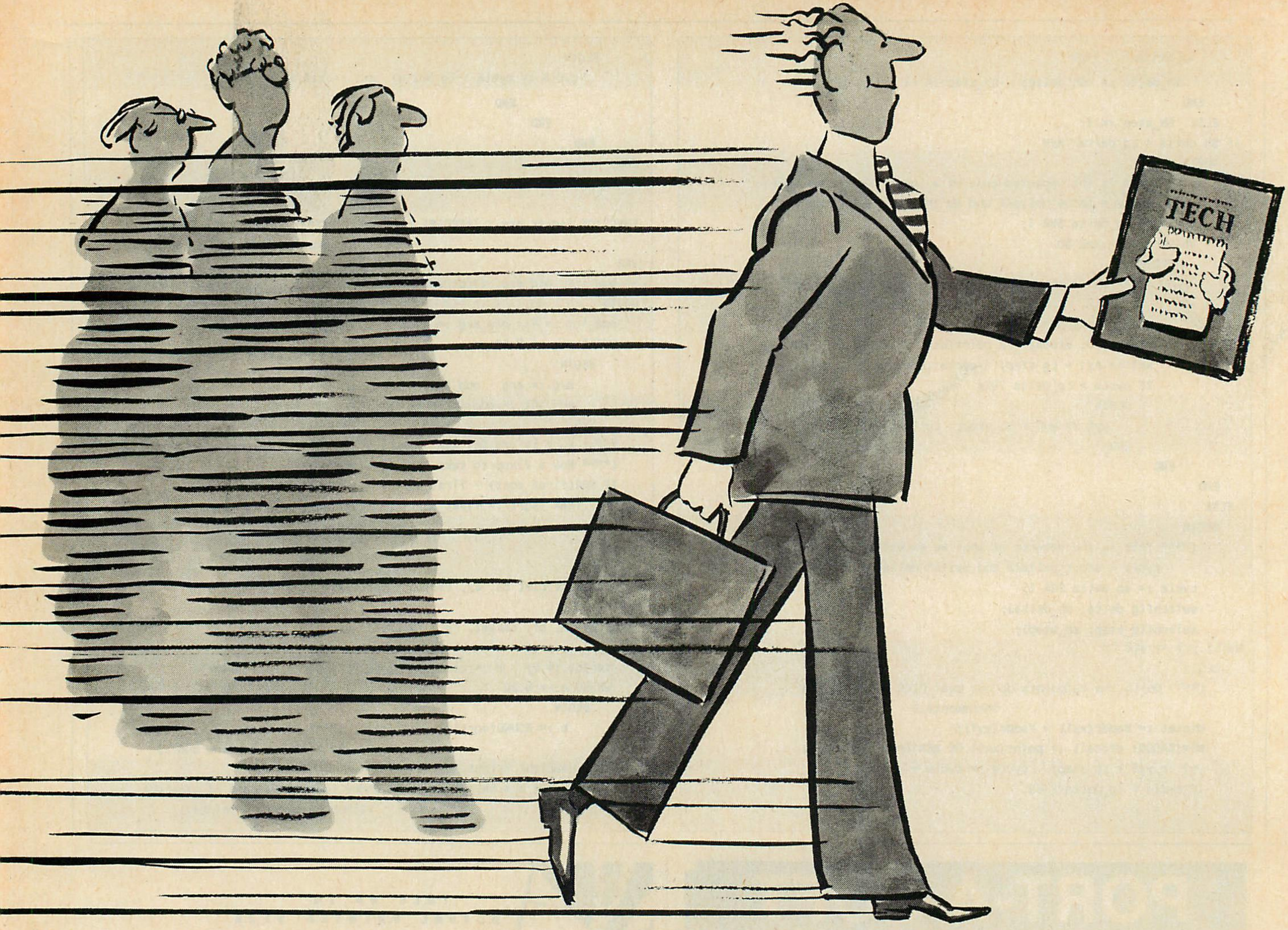
Finally, **screen_dump** prints out the screen on Epson FX, JX, and MX printers with GrafTrax+; the routine supplied with DOS does not support the 640-by-350 mode. **Screen_dump** preserves the screen's pixel proportions so that a circle on the screen comes out as a circle on paper. Minor modifications would permit it to work on other printers. Examples of output are shown in figures 1 and 2.

Victor Mansfield is chairman of the department of physics, associate professor of physics and astronomy, and adjunct professor of computer science at Colgate University.

LISTING 1: EGA_PRIM, INC.

```
{**** An include file that contains EGA Graphics primitives. ****}

TYPE
   RegisterPack = RECORD
         AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags : INTEGER
      END;

VAR
   Rec : RegisterPack;
   Xaddr : ARRAY[0..639] OF INTEGER;
   Yaddr : ARRAY[0..349] OF INTEGER;
   Point : ARRAY[0..639] OF INTEGER;


PROCEDURE Init_graphics;
VAR
   indx     : INTEGER;
   switches : BYTE;
   info     : BYTE;
   mono     : BOOLEAN;
BEGIN
   switches := MEM[$40 : $88] AND $0F;
   info     := MEM[$40 : $87];
   { mono is TRUE if monochrome display attached to EGA }
   mono   := ODD((info AND $02) SHR 1);
   { now we set up registers for the mode set based on }
   { switch settings and information byte bits }
   IF mono THEN rec.ax := $000F  { 640 X 350 monochrome }
   ELSE WITH rec DO
     CASE switches OF
       6 : ax := $0D;   { Color 40 X 25 -- 320 X 200 }
       7 : ax := $0E;   { Color 80 X 25 -- 640 X 200 }
       8 : ax := $10;   { Enhanced color -- normal mode }
       9 : ax := $10;   { Enhanced color -- enhanced mode
       ELSE rec.ax := $0E;  { Default to "safe" 640 X 200 mode}
     END; { CASE }
   INTR($10,rec);
   {**** Arrays used to avoid repetitive address calculations.}
   FOR indx := 0 TO 349 DO Yaddr[indx] := 80*indx;
   FOR indx := 0 TO 639 DO Xaddr[indx] := indx DIV 8;
   FOR indx := 0 TO 639 DO Point[indx] := $80 SHR (indx MOD 8)
END;


PROCEDURE plot( x, y : INTEGER);
VAR
   total : INTEGER;
BEGIN
   total := Xaddr[x] + Yaddr[y];
   {**** EGA memory is sequential starting at $A000.}
   MEM[$A000: total] := point[x] OR MEM[$a000: total]
END;
PROCEDURE switch(VAR first, second : INTEGER);
VAR
   temp : INTEGER;
BEGIN
   temp := first;
   first := second;
   second := temp
END;


PROCEDURE draw( xx1, yy1, xx2, yy2 : INTEGER);
VAR
   Lg_delta, Sh_delta, Cycle, Lg_step, Sh_step, dtotal : INTEGER;
BEGIN
   {**** Set up deltas and steps according to the relationship of
                        (X1, Y1) and (X2, Y2).}
   Lg_delta := xx2 - xx1;   Sh_delta := yy2 - yy1;
   IF Lg_delta < 0 THEN
       BEGIN
          Lg_delta := -Lg_delta;   Lg_step := -1
       END
     ELSE  Lg_step := 1;
```

```
    IF Sh_delta < 0 THEN
        Sh_delta := -Sh_delta;   Sh_step := -1
    END
  ELSE  Sh_step := 1;
IF Sh_delta < Lg_delta THEN
  BEGIN
    {**** Here is the expected case of a longer X-axis so make
          cycle = large_delta/2 and do the normal increments.}
    cycle := lg_delta SHR 1;
    WHILE xx1 <> xx2 DO
      BEGIN
        {**** While the endpoints do not meet Plot and make
                                  the usual increments.}
        dtotal := Xaddr[xx1] + Yaddr[yy1];
        MEM[$A000: dtotal] := point[xx1] OR MEM[$a000: dtotal];
        xx1 := xx1 + Lg_step;   cycle := cycle + Sh_delta;
        IF cycle > Lg_delta THEN
          BEGIN
            yy1 := yy1 + Sh_step;   cycle := cycle - Lg_delta
          END
      END
  END
 ELSE
  BEGIN
    {**** This is the reverse of what we expect so make the
          cycle = short_delta/2 and switch deltas and steps.}
    cycle := sh_delta SHR 1;
    switch(lg_delta, Sh_delta);
    switch(lg_step, sh_step);
WHILE yy1 <> yy2 DO
  BEGIN
   {**** While the endpoints do not meet Plot and make the
                              increments.}
    dtotal := Xaddr[xx1] + Yaddr[yy1];
    MEM[$A000: dtotal] := point[xx1] OR MEM[$A000: dtotal];
    yy1 := yy1 + lg_step;   cycle := cycle + sh_delta;
    IF cycle > lg_delta THEN
      BEGIN
        cycle := cycle - lg_delta;  xx1 := xx1 + sh_step
          END
        END
      END
END;

FUNCTION isqrt( arg : INTEGER) : INTEGER;  {**** Returns the integer
                                 sqrt without using reals.}
VAR
  odd_int, old_arg, first_sqrt : INTEGER;
BEGIN
  odd_int := 1;  old_arg := arg;
  WHILE arg >= 0 DO
    BEGIN
      arg := arg - odd_int;
      odd_int := odd_int + 2;
    END;
  first_sqrt := odd_int SHR 1;
  {**** Now a fixup to take care of overshoots.}
  IF SQR(first_sqrt) - first_sqrt + 1 > old_arg
     THEN isqrt := first_sqrt - 1 ELSE isqrt := first_sqrt
END;

PROCEDURE circle( cx, cy, radius : INTEGER);
VAR
  a, af, b, bf, target, r2 : INTEGER;
BEGIN
  target := 0;   a := radius;   b := 0;   r2 := sqr(radius);
  WHILE a >= b DO
    BEGIN
      b := ROUND(sqrt(r2 - sqr(a))); {**** Use Isqrt(r2 - sqr(a))
                                  here if you have no 8087.}
      switch( target, b);
      WHILE b < target DO {**** Inner loop takes care of straight
                                  lines at cardinal points.}
```


TECH

PROGRAMMER DEVELOPMENT TOOLS

TURBO PASCAL AND UTILITIES

		List	Ours
Turbo PASCAL by Borland International	*Sale*	70	49
Turbo PASCAL *w/8087 or BCD*	*Sale*	110	89
Turbo PASCAL *w/8087 & BCD*	*Sale*	125	99
Btrieve by Softcraft		250	199
FirsTime for Turbo by Spruce Technology		75	69
Multi-Halo Graphics by Media Cybernetics		250	199
Screen Sculptor by Software Bottling		125	109
Turbo ASYNCH by Blaise Computing		100	89
Turbo GRAPHICS TOOLBOX by Borland Int'l.		55	49
Turbo POWER TOOLS by Blaise Computing	*New*	100	89
Turbo TOOLBOX by Borland Int'l		55	49
Turbo TUTOR by Borland Int'l		35	29
TurboPower Util *w/source* by TurboPower Sftwr		95	89
TurboWindow by MetaGraphics		55	49
XTC *Text Editor* by Wendin		99	89

BASIC LANGUAGE

		List	Ours
BetterBASIC by Summit Software		200	169
8087 Math Support		99	89
Run-time Module		250	239
Professional BASIC by Morgan Computing		99	89
8087 Math Support		50	47
True Basic from Addison-Wesley	*New*	150	129
Run-time Module		500	459

OTHER LANGUAGES

		List	Ours
8088 Assembler *w/Z-80 Translator* by 2500 AD		100	89
APL*PLUS/PC by STSC		595	449
Golden Common LISP by Gold Hill		495	Call
Janus/ADA by R&R Software		900	699
MASM-86 *w/utilities* by Microsoft		150	109
Microsoft Fortran	*Links with MS C ver 3*	350	239
Microsoft Pascal	*Links with MS C ver 3*	300	219
Modula-2/86 by Logitech		495	439
Pocket APL by STSC	*with rebate offer*	95	89
Prolog VMI by Automata Design Associates	*New*	100	89
Prolog VML by Automata Design Associates	*New*	300	269
Prolog VMA by Automata Design Associates	*New*	500	439
Prolog-86 by Solution Systems		125	Call
RM/Fortran By Ryan-McFarland		595	439

OTHER LANGUAGE SUPPORT PRODUCTS

		List	Ours
APL2C by Lauer Software	*Interfaces APL to C*	150	139
Btrieve By SoftCraft		250	199
Blaise Tools *for Pascal*		Call	Call
FORTRAN Libraries by Alpha Computer Service		Call	Call
Sci Subroutine Lib *for Fortran or Basic*		175	139

OTHER PRODUCTS

		List	Ours
Advanced Trace-86 by Morgan Computing		175	149
Codesmith-86 *Debugger* by Visual Age		145	129
Polytron Products	*We Carry a Full Line*	Call	Call
Profiler by DWB Associates		125	89
Rtrieve by Softcraft		85	79
Xtrieve by Softcraft		195	169

Periscope Symbolic Debugger
by Data Base Decisions

Write-protect memory board and breakout switch allows instant recovery from runaway code. Provides on-line help, windowing, extensive breakpoints, dual monitor support and more.

List Price $295 Our Price **$269**

PHOENIX PRODUCTS

SUMMERTIME SALE!
In stock and ready for immediate shipping.

	List	Ours
Pasm86 *High Performance Macro Assembler*	295	195
Pfinish *Performance Analyzer*	395	269
Pfix-86 Plus *Symbolic Debugger for Plink-86*	395	269
Plink-86 *Overlay Linker*	395	269
Pmaker *Program Development Manager*	195	119
Pmate *Macro Text Editor*	225	145
Pre-C *Lint Utility*	395	269

Dealer Inquiries Invited

C LANGUAGE

		List	Ours
C-terp *C Interpreter* by Gimpel Software		300	269
Computer Innovations C-86 Compiler		395	299
DeSmet C Compiler with Debugger		159	145
Instant C by Rational Systems	*Sale*	500	399
Lattice C Compiler from Lattice		500	349
Lattice C from Lifeboat	*Ltd Qty Special*	500	275
Mark Williams MWC-86 w/Source Debugger	*Sale*	495	379
Microsoft C Compiler *version 3*	*Sale*	395	259
Wizard C Compiler by Wizard Systems	*Sale*	450	359
Xenix Development System by SCO		1350	1099

MACINTOSH C AND UTILITIES

		List	Ours
c-tree by FairCom	*New version*	395	359
DeSmet/Ouye C Compiler	*New*	150	139
Mac C by Consulair	*Call for Details*	Call	Call
Megamax C *compiler for Macintosh*		295	239

MICROSOFT C AND UTILITIES

		List	Ours
Microsoft C Compiler *version 3*	*Sale*	395	259
Blaise C Tools		125	109
Blaise C Tools 2		100	89
C Power Packs from Software Horizons		Call	Call
C-terp by Gimpel Software		300	269
C Utility Library by Essential Software		185	139
Greenleaf C Functions Library	*New version*	185	139
Greenleaf Comm Library		185	139
The HAMMER by OES Systems	*New*	195	179
Multi-Halo Graphics by Media Cybernetics		250	199
PANEL *Screen Designer* by Roundhill		295	234
Windows for C by Vermont Creative Software		195	139

C UTILITIES

		List	Ours
Basic_C Library by C Source		175	139
Blaise C Tools		125	109
Blaise C Tools 2		100	89
Btrieve by SoftCraft		250	199
C Power Paks From Software Horizons		Call	Call
c-tree by FairCom	*New version*	395	359
C Utility Library by Essential Software		185	139
ESP for C by Bellesoft	*Sale*	349	229
GraphiC by Scientific Endeavors		250	209
Greenleaf C Functions Library	*New Version*	185	139
Greenleaf Comm Library		185	139
MetaWINDOWS by Metagraphics		150	139
Multi-Halo Graphics by Media Cybernetics		250	199
PANEL *Screen Designer* by Roundhill		295	234
PC Lint by Gimpel Software		100	89
Scientific Subroutine Lib *for C* by Peerless		175	139
Windows For C by Vermont Creative Software		195	139

The **HAMMER** C Library by OES
This excellent new C library is designed for creating end-user interfaces. There are functions for creating 123-like menus, managing the screen, creating input forms, prompting for inputs and more. No royalties and includes source code.
List Price $195 Our Price **$179**

LATTICE PRODUCTS

All products receive support and updates directly from Lattice.
Lattice C is in stock and ready for shipment.

	List	Ours
Lattice C Compiler from Lattice	500	349
C-Food Smorgasbord	150	119
C-Sprite *Program Debugger*	175	149
Curses *Screen Manager*	125	109
dBC *C Interface for dBase II or III Files*	250	209
dBC *with source code*	500	439
LMK *Make Facility*	195	159
Text Mgmt Utils *(GREP/DIFF/ED/WC/Extract/Build)*	120	105

TEXT EDITORS

		List	Ours
Brief By Solution Systems		195	Call
Epsilon *Emacs-like Text Editor* by Lugaru		195	179
ESP for C *or* **Pascal** by Bellesoft		Call	Call
ESP for C and Pascal by Bellesoft	*Sale*	399	279
FirsTime for C by Spruce Technology		295	239
FirsTime for MS Pascal by Spruce Technology		245	199
Vedit by Compuview		Call	Call
XTC *Text Editor* by Wendin		99	89

Programmer's Connection

136 Sunnyside Street Hartville, Ohio 44632 (216) 877-3781 (In Ohio)

U.S.: 1-800-336-1166 Canada: 1-800-225-1166

Call For Our Catalog

Account is charged when order is shipped
Prices are subject to change without notice
We accept qualified Corporate Accounts and P.O.'s

```
          BEGIN
            {**** Put in aspect correction and make use of the
                symmetry of a circle by plotting 8 points for
                each calculation.}
            af := 120*a DIV 100;   bf := 120*b DIV 100;
            plot(cx + af, cy + b);  plot(cx + bf, cy + a);
            plot(cx - af, cy + b);  plot(cx - bf, cy + a);
            plot(cx - af, cy - b);  plot(cx - bf, cy - a);
            plot(cx + af, cy - b);  plot(cx + bf, cy - a);
            b := b + 1;
          END;
        a := a - 1;
      END
END;


PROCEDURE Screen dump;
VAR
    xindx, yindx : INTEGER;
    init char : STRING[4];
BEGIN
   {**** The next two lines are for the Epson FX, JX, and MX with
               GrafTrax+.}
   WRITELN(LST,CHR(27)+'A'+CHR(8));
   writeln(lst,chr(27)+'2');
   INIT CHAR := CHR(27)+'K'+CHR(94)+CHR(1);
   FOR xindx := 0 TO 79 DO
     BEGIN
       WRITE(LST, init_char);
       FOR yindx:=349 DOWNTO 0 DO
           WRITE(LST,CHR(MEM[$A000: Yaddr[yindx]+xindx]));
       WRITELN(LST);
       WRITELN(LST,CHR(27) + '@');  {**** Clear printer attributes.}
       WRITELN(LST);
     END
END;
```

LISTING 2: EGA DEMO.PAS-

```
PROGRAM EGA_DEMO; {This program is a simple demonstration of the
EGA graphics primitives found in ega_prim.inc}

{$I ega prim.inc}


VAR
    i, xpos, ypos : INTEGER;
    answer : CHAR;

PROCEDURE triangle(x1, y1, x2, y2, x3, y3 : INTEGER);
BEGIN
   draw(x1, y1, x2, y2);
   draw(x2, y2, x3, y3);
   draw(x3, y3, x1, y1)
END;


BEGIN
   Init graphics;
   FOR i := 1 TO 90 DO
     triangle(0 + i SHL 2, 0, 100, 300 - i SHL 2, 100 + i SHL 2, 50);
   FOR i := 0 TO 250 DO
       BEGIN
          xpos := round( 120*cos( i/pi));
          ypos := round( 100*sin( i/pi));
          circle(320 + xpos, 175 + ypos, i MOD 80)
       END;
   FOR i := 5 TO 65 DO circle(585, 300 - i SHL 2, 35);
   {**** No prompt is given the user since it would be printed.}
   READ(kbd, answer);  IF upcase(answer) = 'Y' THEN Screen dump;
   textmode(bw80)
END.
```

"I Program In BetterBASIC And I Recommend It."

"My company uses BetterBASIC to control the automation system on board ships. It gives us the excellent feature of flexibility and quick change of process dependant logic. We save the logic in E²prom (electrically erasable programmable ROM) so online changes can be made instantly. I use BetterBASIC and I recommend it."

Ken Moller · Norcontrol · Horten, Norway

640K Now you can use the full memory of your PC to develop large programs.
STRUCTURED Create well organized programs using procedures and functions that are easily identified and understood and completely reusable in future programs.
MODULAR Use procedures and functions grouped together to form "library modules" which are then available to you or anyone else for future use.
EXTENSIBLE Create your own BetterBASIC modules which contain BetterBASIC extensions. This feature coupled with the easy-to-use Assembly Language support, makes this an ideal OEM language.
INTERACTIVE BetterBASIC acts like an interpreter because it responds to the users' commands in an immediate mode. However each statement is actually compiled as it is entered.
COMPILED Each line of the program is compiled as it is entered into the computer's memory rather than interpreted at runtime.
RUNTIME SYSTEM The optional Runtime System generates stand alone EXE. files allowing for the distribution of products written in BetterBASIC with no royalties.
SUPPORTS Windows, Graphics, DOS and BIOS ROM calls, Chaining, Overlays, Local and Global Variables, Recursion ... and more.
BetterBASIC Runs on IBM PC, XT, AT and all IBM-compatibles. Ask your local dealer for BetterBASIC or call 1-800-225-5800 in Canada call 416-469-5244. Also available for the Tandy 1000, 1200, AND 2000 AT Tandy/Radio Shack stores.

Summit Software Technology, Inc.™
P.O. Box 99, Babson Park
Wellesley, MA 02157

PRICES:

BetterBASIC	$199
8087/80287 Math Module	$99
BTrieve™ Interface	$99
Runtime System	$250
Sample Disk with Tutorial	$10

Better BASIC.

Because It's The Best.

CIRCLE NO. 179 ON READER SERVICE CARD

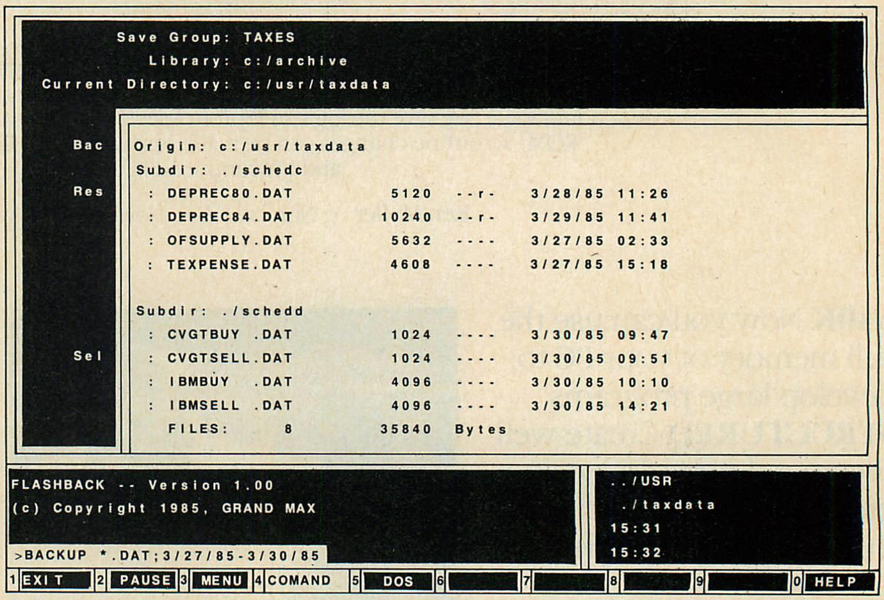
MasterCard, Visa, Checks, Money Order, C.O.D. accepted and P.O. on approval.

BetterBASIC is a registered trademark of Summit Software Technology Inc. IBM PC, XT, AT, are registered trademarks of International Business Machines Corp. Tandy is a registered trademark of Tandy Corp. Btrieve is a registered trademark of SoftCraft Inc.

(If you're using BetterBASIC and would like to be featured in one of our ads, please write to the Director of Advertising at Summit.)

FILE MANAGEMENT is
Fast Backup and much, much more...

FILEMASTER supports a versatile file selection system which lets you process files by name key, modification date range, or name key range, using relative or absolute path specifications. . .



FILEMASTER

The definitive backup and file management system for hard disk users.

FILEMASTER is a complete set of tools for managing the complexities of the DOS file system. In one integrated package, you get seven powerful file management utilities:

FLASHBACK; over twice as fast as the standard DOS backup utility, with options for data compression, file encryption, data validation, and on-line directories of file archives stored on diskettes. Once a save set is created, FLASHBACK will automatically save only those files which have been changed or added since the previous save.

Using sophisticated data compression techniques, FLASHBACK saves storage space for file backups, and lets you create compact, encrypted data packets for use in computer-to-computer data transfers.

BDIR; gives you a quick directory listing of files stored off-line in your backup archives.

FINDEX; an application driver which lets you select and process multiple file sets anywhere on your disk performing any of several standard functions, or linking with applications of your own. Once FINDEX starts, it works automatically, crossing data paths without user intervention.

TEXTED; a global string search & display, search & modify utility which allows you to make large scale changes to files on your disk, both binary and ascii.

TUNEUP; a hard disk tune-up process which sorts directories, frees up wasted directory space, and optimizes disk storage to make your hard disk run smoother and faster.

DISCOPE; a powerful storage analysis program which allows you to examine, compare, and modify disk sectors, either as part of a file chain, or in arbitrary combinations.

EXPATH; a path extension utility which allows you to reference files in different directories by name without changing your current directory.

You've probably seen similar utilities advertised individually before, and at prices well over $100 each. With FILEMASTER, you get them all for only:

$49.95

including a complete user manual & binder, contemporary multi-window, pop-up menu interface, and a support line for technical assistance.

Hardware Requirements: IBM-PC XT, Compaq-Plus, or compatible equipment; DOS version 2.xx; minimum 128K, 256K recommended for optimum performance.

FILEMASTER is available only from Grand Max Software Corporation. To order with VISA or Mastercard, dial:

1-800-624-3285

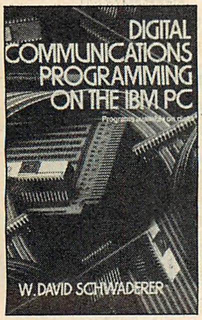
In Oregon: (503) 635-8295

Grand Max
SOFTWARE CORPORATION
68 LEONARD STREET • LAKE OSWEGO, OR 97034

Communications Basics

Beginning and intermediate PC users interested in digital communications will find this book valuable.

Digital Communications Programming on the IBM PC
W. David Schwaderer (John Wiley & Sons, New York, NY, 1984) 332 pages; $17.95 (with program disk, $47.90)



The popularity of PCs and the recognition that connecting them to one another and to mainframe systems is beneficial has spawned a whole new subindustry—microcomputer communications. Stated simply, such communications involve sending data between terminals and computers over dedicated and switched telephone lines. Digital communications are difficult to explain, even to experienced programmers; explaining them to persons who are new to both communications and programming, as *Digital Communications Programming on the IBM PC* is intended to do, is a significant challenge.

W. David Schwaderer has written several communications programs for the IBM PC and has committed to paper and disk the concepts, applications, and special considerations involved in making data connections. In this book, he starts by specifying what a digital communications system should be able to do, then takes the reader through the design and coding of various communications programs, including a working computer bulletin-board system. He also examines the COMM.BAS program provided with PC-DOS.

The book presents topics clearly and simply and offers numerous illustrations to explain key points and established relationships among the diverse elements of communications hardware and software. The broad scope of topics covered, coupled with a limited but sufficient depth to permit useful work to be accomplished, create a well-balanced book that could be important for many PC users.

By the author's own admission, C is probably a better language for systems and applications software, but all examples in the book are written in BASIC. Interpreted programs are used, but helpful information is provided in a separate chapter for users of the BASIC compiler and the Microsoft linker. BASIC was chosen because it is the only language available with all IBM PCs and because the Microsoft version contains extensive support functions and statements for accessing the communications ports and using event traps.

The book has five major sections dealing with communications concepts, key components, BASIC language support functions, programming considerations and examples, and additional topics. There are 16 chapters in all, plus 8 appendixes, several of which reproduce almost verbatim information in the *IBM BASIC* and *Technical Reference* manuals. Thus, the essential programming and hardware information is available in one convenient package.

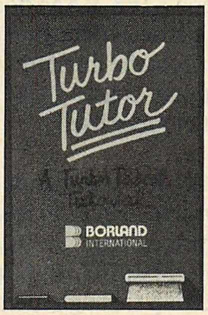
The programs in the book are available on a companion disk that may be purchased separately or with the book. These programs, 78 of them, are written for the IBM PC, PC/XT, and PC*jr* (Appendix H describes needed modifications for programs to be run on a PC*jr*). In addition, the programs appear to work properly on the PC/AT.

Both the programs and the text provide good coverage of keyboard and screen interactions, event trapping, and interrupt handling. The material on event trapping is especially detailed and useful, because little information has been published on this topic.

More advanced PC users would probably prefer more in-depth coverage of the topics and examples provided in assembly language or some other language more suited to systems-level programming tasks than BASIC. *Digital Communications Programming on the IBM PC* can be recommended, however, as a good introduction to beginning and intermediate computer users wishing to tune in on what is happening in the world of digital communications.

—AUGIE HANSEN

Turbo Tutor
(Borland International, Scotts Valley, CA 1985) 270 pages; softcover; $34.95 with program diskette



Turbo Pascal, the best-selling compiler from Borland International, is fast becoming the "people's compiler" because of its low price, excellent performance, and full line of features.

Many of the program's purchasers are first-time Pascal users who have little programming experience. Consequently, Borland decided to produce *Turbo Tutor: A Turbo Pascal Tutorial* to teach this multitude of new programmers how to use the compiler and how to program in Turbo Pascal. The accuracy of Borland's claim that *Turbo Tutor* may be the only reference on Pascal and programming that a user may ever need depends, of course, upon his background, programming experience, and intuition.

Turbo Tutor is a book about Turbo Pascal with an accompanying diskette rather than a diskette-based tutorial. It demands an active interest on the part of the reader because it does not provide interactive instruction: *Turbo Tutor* does not teach the user Turbo Pascal, it helps him to learn.

The book is divided into three major sections: "Turbo Pascal for the Absolute Novice," "A Programmer's Guide to Turbo Pascal," and "Advanced Topics in

Turbo Pascal." The book's intention is to indoctrinate the user who is completely unfamiliar with Pascal by providing the total picture, while its expectation is that most readers will have some computer/programming experience and will therefore begin with the second section, progressing to the final part as they sharpen their skills.

The novice reader is treated to a short history of programming, followed by a discussion of writing and developing a program. The information on getting started with Turbo Pascal is the most important part of the first section.

The real meat of *Turbo Tutor*, however, is contained in the second part, "A Programmer's Guide to Turbo Pascal," which covers everything from the basics of Pascal to arrays, pointers, and files. Many of the examples are related to an old computer game called *Star Trek*, making an understanding of that game helpful in fully appreciating the discussion. Fortunately, each example is explained in detail, but a more varied sampling would have been helpful.

Most of the third section is made up of commented listings of Turbo Pascal programs. No explanations are given apart from the internal comments, but if the reader has fully understood the previous sections, this should not be a problem. Three generic Turbo Pascal routines are included, along with 10 MS-DOS, two CP/M, and two assembly language routines. The listings provide some insight into the power and features of Turbo Pascal; the procedures are suitable for copying into other programs and for use as examples when the reader writes his own routines.

The book's style is conversational and easy to understand. Its chapters are short (the longest is 26 pages) and stick to their stated topics. The sequence of subjects is logical and each chapter builds on what has already been stated. The program diskette consists of the source files for the programs in part three and the source code for a large Turbo program called ListT. (ListT provides formatted listings of Turbo Pascal source code; it is quite informative.)

Turbo Tutor has something for everyone, from the novice to the experienced Pascal devotee, and it is reasonably priced in relation to other book/program diskette combinations (the diskette is not optional). With some effort, a novice can learn Turbo Pascal from this book, but it probably is not the only reference on Turbo that a user would ever need.

—BRUCE T. FILBECK

BRIEF™

"BRIEF Is The Best Editor I Have Ever Used. I Switched. I Use BRIEF Regularly."

Steve McMahon*

Choosing the right program editor is an extremely important decision. After all, more time is spent with your program editor than with any other category of software. A good program editor helps to stimulate creativity and produce programs of superior design. A poor program editor, on the other hand, can be worse than none at all. It can slow you down, get in your way, and make even the simplest tasks difficult.

BRIEF - PROGRAM EDITING *YOUR WAY*

Every programmer has an individual style that makes their work unique. Most program editors compromise that style by forcing the programmer to conform to their methods. Not so with BRIEF.

BRIEF's most powerful feature is its ability to conform to your way of programming. BRIEF can be easily tailored/customized to your individual preferences. For example, keyboard reassignment allows the keyboard to be used in whatever way you prefer. Keystroke macros allow long sequences of editor directives to be repeated by pressing a single key. Plus you can simultaneously work with numerous program and data files, and configure many windows to control BRIEF's visual presentation.

75% OF THE EXPERTS WHO HAVE TRIED BRIEF SWITCHED!

(Call For Details)

"A BONA FIDE UNDO" Steve McMahon – BYTE REVIEW "BUILD YOUR DREAM EDITOR" MARCH 1985

Here are Steve McMahon's exact words: "...BRIEF implements a true undo facility, by default allowing command-by-command recovery from the last 30 commands...The number of commands you want to be able to undo can be changed." (up to 300) "Only with BRIEF, though, was it possible to undo a macro that produced 4000 words of text with a single keystroke."

EVERY FEATURE YOU WANT

BRIEF supports practically every feature you've ever seen, and some you probably haven't thought of yet.

- Edit Multiple Large Files
- Windows (Tiled & Pop-up)
- Unlimited File Size
- Full UNDO (N Times)
- True Automatic Indent for C
- Exit to DOS Inside BRIEF
- Compile programs inside BRIEF
- Uses All Available Memory
- Intuitive Commands
- Tutorial
- Repeat Keystroke Sequences
- 15 Minute Learning
- Reconfigurable Keyboard
- Online Help
- Search for "Regular Expressions"
- Mnemonic Key Assignments
- Horizontal Scrolling
- Comprehensive Error Recovery
- And ... a Complete, Compiled, Programmable and Readable Macro Language.

```
make.c
int       handle = 0;

main (argc, argv

int              argc;

fsa.h
#include "..\include\ctype.h"

typedef  struct
{
   short     action,
             state;
} Fsa;
#ifdef    FSA_MAIN
Fsa       fsa[][8] = {/* Alphanum    Co

/* State 0. */            0, 2,      10
/* State 1. */            10, 0,     10
/* State 2. */            0, 2,      1,
/* State 3. */            0, 5,      11
/* State 4. */            0, 4,      0,

makefile.h
/**
 **              makefile.h:
 **
 **         This is the definitions fil
 **   Hopefully, it won't be unreasonab
 **   that have been written.
 **/

typedef struct cmd_struct
{
   char        *cmd_text;

   struct cmd_struct *next_cmd;
} *Cmd_Ptr, Cmd;

Mismatched open parenthesis.          Line: 11   Col: 17      2:17 pm
```

A TYPICAL BRIEF SCREEN

Notice there are three windows on the screen simultaneously and each one is viewing a different file. The mainline of a C program is visible in the uppermost window; the programmer has run a syntax checking macro which found a mismatched open parenthesis in the arguments to the mainline. The other two windows show header files containing information crucial to the design of the program. BRIEF can have an unlimited number of windows and files accessed simultaneously.

BRIEF'S ONLY LIMITATIONS ARE THE ONES *YOU* SET

Full refund if not satisfied in 30 days. ONLY $195

→ **FREE** with order: "Best of BRIEF Macros" — Contest entries include macros for Fortran, C, Calculator, Base Converter, etc. Call before November 30. **800-821-2492**

AVAILABILITY: IBM-PC and Compatibles, AT, & TANDY 2000

Solution Systems™

335-P Washington St., Norwell, MA 02061 (617) 659-1571 **800-821-2492.**

Steve McMahon's quote courtesy Suntype Publishing Systems. BYTE review by Mr. McMahon may be found in BYTE Magazine March 1985.

Shrink-wrapped Enforcement

Legal protection for software publishers is bringing the issue of user's rights into focus.

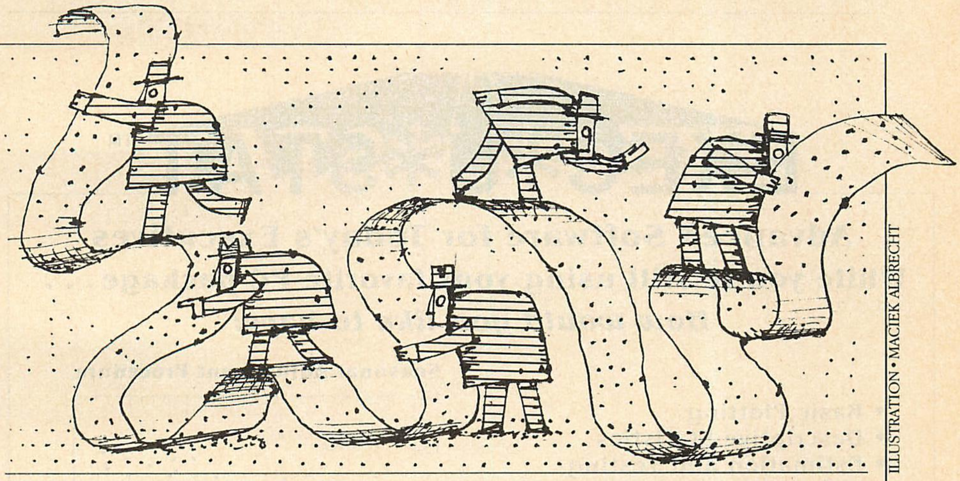


ILLUSTRATION • MACIEK ALBRECHT

Why would a California software company travel to Louisiana to sue a Canadian software company? To try to improve its chances of enforcing a "shrink-wrapped" license.

The first Legal Brief ("The IBM Software License," July/August 1983, p. 149) noted the uncertainty of enforcing a shrink-wrapped license in a transaction that otherwise looked very much like a plain vanilla sale. More than two years later the same uncertainty exists. In an effort to eliminate that uncertainty, the Louisiana state legislature passed the Software License Enforcement Act (SLEA), which provides that in certain circumstances a purchaser of software is deemed to have accepted all provisions of a shrink-wrapped license. Other states are considering the adoption of SLEAs.

Vault Corporation vs. Quaid Ltd. is the first case to test the validity of such a statute. The California-based Vault Corporation alleges that Quaid, the Canadian producer of software that makes back-up copies of copy-protected software, decompiled a program in violation of a shrink-wrapped license agreement. It does not allege that the program was produced, purchased, or decompiled in Louisiana, so Vault vs. Quaid presents a less-than-perfect case for putting such a law to the test. If, in fact, no such contact occurred with Louisiana, the situation would be akin to the Canadian government deciding to set the maximum speed limit in the state of Louisiana.

The Louisiana SLEA (R.S. 51: 1961-1966) provides that any person who acquires computer software is "conclusively deemed to have accepted and agreed to all the terms of the license agreement for such software" if a prominently displayed, readily understandable legend spelled out in capital letters is conspicuously visible on the software packaging; the legend must state clearly that use of the software constitutes acceptance of the license terms or that opening a sealed package constitutes acceptance of the license terms. (One solution may be to have a minor take the "acceptance" action, whatever that action may be; although the minor might have accepted the terms of the license agreement in doing so, it is arguably unenforceable against him.)

The only substantive preconditions to the statute are the requirements that the legend must provide that the purchaser may, if he does not accept the license terms, return the unused, unopened software for a full refund, and that the word *license*, "either alone or in combination with other words, appears prominently at or near the top of such document." A strictly technical interpretation of this precondition would mean that the heading "YOU DON'T NEED TO READ THE LICENSE AGREEMENT" would comply. (The statute also provides that the license may not be concealed inside the sealed package.)

Some uncertainty still surrounds the Louisiana statute. It is neither intended to limit the effectiveness of any license agreement under any other Louisiana law, nor is it intended "to limit in any manner the laws of this state or any other laws." Presumably, the Louisiana legislature was concerned about the validity of a statute in conflict with federal copyright law (current copyright law permits owners of software to make archival copies, but it does not prohibit them from forfeiting that right by contract, although there may be a point at which protection schemes are so at variance with the purpose of the copyright law that copyright is forfeited) or inconsistent with the commerce, comity, or full-faith and credit clauses of the U.S. Constitution.

UNCERTAIN CONSEQUENCES

Troubling issues are raised by the concept of a statute that resolves the uncertainty of unilateral software contracts (contracts in which one party makes an offer that the second party may transform into a contract by performance) so one-sidedly. Implicit in such a statute are three propositions:

1. Software users actually read and understand the license agreements. (The Louisiana act requires that the *notice* that a license agreement lurks within must be "in language which is readily understandable to a person of average literacy," not, however, that the license agreement itself must be.)
2. Software publishers need special protection, different from publishers of books or manufacturers of automobiles. As a corollary, it must be

the end users whom the publishers need to be protected from.

3. The converse of proposition 2 is not true; end users need no special protection.

These propositions raise some interesting questions. Judging from inquiries that are received from sophisticated purchasers, I doubt that the average purchaser of a computer program understands all aspects of the license agreement. For example, the distinction between consequential and incidental damages is one that often eludes even lawyers (it generally makes no practical difference because most license agreements disclaim both of them).

What is the rationale for granting software publishers special statutory rights? Tom Mack, president of the Capital PC Users Group, offers an interesting comparison: if American automobile manufacturers were entitled to similar protection, a purchaser of a shrink-wrapped new car could find that by removing the wrapping he has agreed that there is no warranty, that only he may drive the car, and that he cannot resell it. (The Louisiana law lists several enforceable licensing terms, including a provision prohibiting transfer except by operation of law or in connection with the sale of substantially all of the assets of the licensee's business. One way around such a provision would be to set up separate corporations to hold each license.)

Proposition 3 is probably the most dangerous. The Louisiana law suggests several acceptable license provisions (mostly dealing with ownership and transferability), but it also provides that the purchaser has agreed to *all* of the license terms. Inevitably, the standard disclaimer of liability and disclaimers of warranty will be included.

An unscrupulous company could sell blank diskettes that were colorfuly packaged and highly promoted as combination word processor, spreadsheet, database management programs—and sell them quite inexpensively, thanks to the license agreement disclaiming any warranty or liability (or setting a statute of limitations on claims at, say, one minute). By the time the unwary purchaser tried the disk and found it blank he would already be bound by the license agreement and without remedy in the situation.

UNCLEAR RESTRICTIONS

Drawing from the statute itself, section 1964(3) explicitly approves prohibition of "reverse engineering, decompiling, or disassembling," which is defined in section 1962(3) as "any process by which computer software is converted from one form to another form which is more readily understandable to human beings, including without limitation any decoding or decrypting of any computer program which has been encoded or encrypted in any manner." When a disk is booted and produces user-friendly output, is that not arguably within the statutory definition of reverse engineering, decompiling, or disassembling? What about the creation of a spreadsheet? The execution of object code? Did the Louisiana legislature actually intend to allow the sale of computer software that the user is prohibited, by this definition, from using?

Other states are currently considering SLEAs. If such statutes are necessary, at the very least a balance should be reached. Any company that is seeking the protection of an SLEA should be required to grant minimal license terms to the buyer in return.

Large corporate users generally negotiate their own license agreements and therefore will be relatively unaffected by shrink-wrapped licenses. Effective advocacy for a balanced statute, therefore, must come from the individual users. One attempt already under way is the Bill of Rights for Software Users, developed by the Capital PC User's Group and reprinted in full in this month's editorial. (See "User's Rights," Will Fastie, p. 9.)

APPROACHING A BALANCE

Legislation that is philosophically similar to the position being promoted by CPCUG is pending in California. As introduced, however, the bill does not impose the entire Bill of Rights, but does evidence a concern for consumer interests in computer software (and hardware) purchases.

The bill requires all sales or leases of new computer hardware and software in California (recognizing the limits of sovereignty that are overlooked by Louisiana) to be accompanied by minimum warranties:

1. Merchantability (if the product is sold by a dealer), which is defined as including conformity to the product description as used in the trade as well as any promises made on its label, fitness for the ordinary purposes to which the class is used, and adequate labelling.
2. Fitness for the buyer's purpose (if the manufacturer or seller knows that the buyer is relying on the manufacturer's or seller's skill or judgment to select the appropriate product).
3. Conformity to advertised claims by the manufacturer.
4. Conformity to advertised claims by the seller.

The bill has strong remedy provisions as well. Where a manufacturer or seller has provided that the buyer's remedies are limited and the limited remedies do not cure a breach of warranty, the buyer may revoke acceptance of the product within a reasonable time (which is defined to be at least six months after purchase or installation). Under the Uniform Commercial Code, this would give the buyer the right to return the product and receive a refund. If the seller willfully refuses to honor the buyer's right, the buyer *may* (that is, the Court has the power but not the duty to allow the buyer to) recover punitive damages as well as actual damages. Perhaps most signficantly, if a buyer prevails he shall (particularly when used in the same statute that also uses the term may, usually interpreted as mandatory) be awarded reasonable attorney's fees and other reasonable litigation expenses. The term reasonable does not necessarily mean negligible.

The California bill can be criticized as overly consumer-oriented. It is not limited to shrink-wrapped licenses or to consumer sales. In addition, it may not be appropriate to impose specific warranties where both parties to the transaction have equivalent bargaining power and information.

A balance certainly exists somewhere between the approaches being taken by the Louisiana and California legislatures. Until that balance is achieved, Louisiana software users might consider buying software through the mail, thereby raising the issue of whether the Louisiana statute reaches interstate transactions.

Max Stul Oppenheimer, PC, is a partner in the law firm of Venable, Baetjer & Howard, located in Baltimore, Maryland.

TECH BOOK

ACCESSORIES/SUPPLIES

RESET BUTTON FOR IBM PC
The button IBM forgot. A good reset button is a basic tool for PC (PC/XT) programmers. When your machine is hung up, a touch of the button will quickly reboot your computer. This kit installs easily and mounts out of the way behind your machine. Easy instructions included. Send $19.80 (add 6% in MN).
RESET CORPORATION
P.O. Box 14809
Minneapolis, Minnesota 55414
(612) 489-5892

BUS. OPPORTUNITIES

EXTRA INCOME
.. Learn 3 steps to Profitable Consulting
.... How to Locate Prospects
.... How to Bid on Projects
.... How to Reference Sell New Clients
.. Discover 4 Profitable Client Services
Invest $9.95 in "Make Your PC Profitable."
FREE Table of Contents sent on request.
TULSA COMPUTER CONSORTIUM (TCC)
P.O. Box 707
Owasso, OK 74055
918-747-0151

HARDWARE/ADD-ON BOARDS

LOW COST MODULAR DAS SYSTEM
Now available Modular INSTRUMENTAL DATA ACQUISITION SYSTEM (MIDAS) for IBM PC family. Hardware provides direct connection to standard transducers. SIGNAL CONDITIONING INCLUDED. Software provides PC DOS drivers for compatibility with other application programs. Features include 10w unit cost, high accuracy, modularity, and expandibility.
ADVANCED PERIPHERALS INCORPORATED
12650 West Geauga Plaza
Chesterland, OH 44026
(216) 729-3927

DT2801 SERIES ANALOG I/O
Plug-in data acquisition boards with 8DI/16SE analog inputs, high or low level programmable gains, 2 analog outputs, 16 lines of digital I/O, DMA, on-board clock, and on-board microprocessor. Optional software subroutines and screw terminal panels.
DATA TRANSLATION INC.
100 Locke Drive
Marlboro, MA 01752
(617) 481-3700

$200 PER MEGABYTE
Memory expansion for your IBM PC, XT, OR AT at $200 per megabyte. All the memory you've dreamed of at a price you can live with. JRAM-2 with JRAM-AT hold up to 2 megabytes per board with Ser/Par/Clks also available. Mega-memory and I/O ports all in one slot!
Tall Tree Systems
1120 San Antonio Road
Palo Alto, CA 94303
(415) 964-1980

FIXED DISK BIOS/BOOT
FiXT boots from most popular Hard Disks—DAVONG, TECMAR, IOMEGA, GT LAKES, etc. Adds XT-like BIOS interface to your disk for PC. Security, multiple volumes, removable media support optional. No-slot plug-in installation. Specify controller and computer with order. $80-$95. Add $3 shpg, CA tax.
GOLDEN BOW SYSTEMS
P.O. BOX 3039
SAN DIEGO, CA 92103
(619) 298-9349

HARDWARE/PERIPHERALS

A/D TO RS232 CONVERTER
Low cost stand-alone data acquisition board for any 8 bit RS 232 port. 8 channels, 8 bit conversion. 0-5V input, 300-9600 baud. Small size, 4" × 5", with wall plug power supply. Easy-to-use, includes sample driven software listings. Kit w/all parts & PS, $59.95 + $4 shipping. Assembled & tested is $89.95 + $4.
TPS SYSTEMS
14820 Elmore Road
Anchorage, AK 99516
(907) 345-6730

BARCODE READERS
- IBM PC/XT, AT keyboard or RS-232 interface—
- NO programming required for IBM PC/XT, AT keyboard interface unit.
- Reads dot-matrix & printed labels accurately
- Choice of digital wands
- Reads UPC A/E, code 39, codabar & I 2 of 5
- $595/$635 list.

PERCON
2190 West 11
Eugene, OR 97402
503-344-1189

33% FASTER AT—$19.95
If you have a PC AT you can plug in a new crystal, and watch it run 1/3 faster. Non-technical 10 minute installation (no soldering), and no expansion slot usage. Compatible with all software. Satisfaction guaranteed. Check or Money order. COD. $3 extra.
AVTEK ENGINEERING
P.O. Box 21-171
Eagan, MN 55121
(612) 452-7415

HARDWARE COMMUNICATION

THE VISIONARY 1200
The Visionary Electronics super-smart modem sends and receives messages anytime, day or night, with or without a computer. Stores files, names, phone numbers, and more in RAM; complete communications software in ROM; on-board realtime clock; 8085 driven; NiCad memory back-up; and TELEX terminal emulation.
Visionary Electronics Incorporated
VISIONARY ELECTRONICS
141 Parker Ave.
San Francisco, CA 94118
(415) 751-8811

HARDWARE/DISK DRIVES

1.2MB PC-AT DISK DRIVES
AT your PC by replacing your diskette controller and adding a high-density disk drive. JDISKETTE handles 1 to 4 diskette drives plus and I/O module. $249 for JDISKETTE-SA, $250 for TEAC 55GF (1.2MB); $200 for TEAC 55B (36-K); $180 for Ser/Par/Clk module.
TALL TREE SYSTEMS
1120 San Antonio Rd
Palo Alto, CA 94303
(415) 964-1980

SOFTWARE/ARTIFICIAL INTELLIGENCE

A.D.A. A.I. PROLOG-LISP TOOLKIT
A.I. Toolkit. A.D.A. PROLOG and Cybermetrics. UNX LISP in one package. Virtual memory PROLOG has full C&M syntax. LISP has macros, catch, throw, optional and rest parameters, structure editor and utilities w/source. Both systems have tree structured object list. Use with any text editor. $149.95
A.D.A.
1570 Arran Way
Dresher, PA 19025
215-646-4894

SOFTWARE/BUSINESS OPPORTUNITIES

RETRACTION
Below is the CORRECTED version of an advertisement which appeared in our May June July issues

SOFTWARE AUTHORS
WE WILL CONNECT YOU TO MAJOR SOFTWARE PUBLISHERS FREE. We are in contact with many publishers looking for marketable software. If you have a quality software package or program and wish to have it published and distributed, contact:
SOFTSEARCH, Inc.
P.O. Box 281
Budd Lake, NJ 07828
(201) 627-1790

VERTICAL MARKET OPPORTUNITY
If you develop software for the Advertising, Publishing or Broadcast industries call us. We're Interactive Market Systems. We have 10,000 clients in those industries World-Wide. If your product can benefit our client base, we'll provide marketing and sales support. If you currently have a product or are about to develop one give us a call. Contact Tami Peterson.
IMS Micros
355 Lexington Avenue
New York, N.Y. 10017
212-682-9494 1-800-223-7942

SOFTWARE/COMMUNICATIONS

FoReM PC Bulletin Board System
Advanced Bulletin Board System for IBM PC. Multiple message bases; Electronic Mail; Xmodem; 255 access levels; Remote sysop access; Privilege system; Hard disk support; Menu driven text file Data Base ... Much more!
Commnet Systems
7348 Green Oak Terrace
Lanham, MD 20706
(301) 552-2517

RATES AND INFORMATION

Standard listings consist of a bold lead line (23 characters maximum); 7 lines of body copy (45 characters per line); 4 lines for company name, address and telephone number. $110 per insertion—3 issue minimum. Additional charge for extra lines and company logos. Prepayment and frequency discounts available. American Express, MasterCard, Visa accepted. Copy subject to publishers approval. Send typewritten or printed copy, reproducible logo art (if applicable) and remittance to Kathryn Cumberlander, Classified Sales Manager, Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. For additional information, assistance, or to place an order by phone, call collect (212) 503-5115.

TECH BOOK

SOFTWARE /COMPILER

DeSmet-C - $109
Full K&R C compiler, linker, librarian, fullscreen editor, profiler, overlays, example software, 140 page manual. BOTH 8087 and floating point libraries. OUTSTANDING PRICE/PERFORMANCE. Consistently rated 1st or 2nd in published benchmarks. PC/MSDOS. No royalties on generated code. Newsletter. Updates $20 ea. $109 shipping included. Symbolic (C source level). Debugger Option $50 extra. C Compiler for the Macintosh $150.
C WARE CORPORATION
Dept. PCTJ
P.O. Box C
Sunnyvale, CA 94087
(408) 720-9696

C LIBRARIES + C WINDOWS
***** BEST YOU CAN GET *****
Over 325 richly commented tested functions
ALL SOURCE CODE! .... NO ROYALTIES!
*** 4 DISKS: 400 PAGE MANUAL ***
51 screen handling/graphics funct. $49.95
50 cursor/keyboard/data input $39.95
85 superior string functions $59.95
28 system status & control funct's $29.95
70 utilities/DOS/BIOS/time/date $59.95
42 printer control functions $39.95
No matter what ELSE you have, get these!!
Any 3 libraries..... $69.95 All 6......... $99.95

C POWER WINDOWS C POWER WINDOWS
PROFESSIONAL WINDOW MANAGEMENT
overlays, borders, pop-up menus/help windows
status-line, color, highlighting, much more!
C WINDOWS: C complete source code .. $99.95

* SUPERFONTS FOR C: Dramatic, high impact
* screens. Use our font and image libraries
* or create your own. Use with/without Windows.

C windows' plus 6 libraries $159.95
* plus SUPERFONTS FOR C, all 3 for $199.95

Entelekon
SOFTWARE SYSTEMS

Entelekon
12118 Kimberley
Houston, TX 77024
(713) 468-4412

SOFTWARE DATA ENTRY

Data Entry System
Heads down data entry for the PC will allow fast and accurate preparation of data files for your mainframe. Features: Auto dup, verification, auto format change, table look-ups, range checks, batch totals, dup key, record insert, and much more. Ask us about our free 30 day trial period. Only $395.
COMPUTER KEYES
6519 193 SW
Lynnwood, WA 98036
(206) 776-6443

Semi-Custom® DATA ENTRY
Users frustrated by manuals, codes or forms? General purpose data entry software doesn't suit you? Add a front-end tailored to your specifications. Fully supported turn-key data entry module designed for occasional users. Fixed price. $1,000-$7,000. Users manual on a Business card! For free quote & sample, call:
SEMI-CUSTOM ® SOFTWARE
6807 Brennon Lane, Suite 110
Chevy Chase, MD 20815
(301) 654-8135

SOFTWARE DEVELOPMENT TOOLS

THE FORMS DESIGNER™
Attention IBM Pascal, FORTRAN users! Save time in designing formatted screen I/O. Interactive Forms Editor allows you to draw lines and boxes, define fields, and edit text. Access forms or read keyboard entry by writing only one line of code. Provides sequential data retrieval and storage. Requires 128K RAM. Only $275 complete. Demo and manual $35. Call or write:
BIT SOFTWARE
PO Box 619
Milpitas, CA 95035
(408) 262-1054

TURBO-PLUS™
Turbo-Plus™ is a set of external procedures crafted in assembler language and designed specifically for Turbo Pascal's interactive environment. Turbo-Plus™ includes: File Handle Disk Access Method, Instant Memory Mapped Text Display, Expanded Text Display. $34.95 plus $5 s&h.
NOSTRADAMUS
5320 South 900 East
Salt Lake City, UT 84121
(801) 261-0769

ASSEMBLE 3-6 TIMES FASTER
"FAST ASSEM 86" is much faster and easier to use than the IBM Assembler. Assembles 8K files 6 times faster. Editor, assembler and source always in RAM so you can go instantly from editing to assembling. No need to LINK since produces .COM (as well as .OBJ) files directly. Source compatible with IBM's important features. Supports 186, 286, 8087, 287 directly. X-ref. utility built in. Only $49, with .OBJ $99.
SPEEDWARE
118 Buck Circle
Folsom, CA 95630
(916) 988-7426

COLOR, CURSOR CONTROL
DATE/TIME, KEYBOARD, PEEK, POKE, COMMAND LINE ACCESS, and more available for IBM/MS FORTRAN, IBM/MS PASCAL, and GENERIC (MS) BASIC on PC/DOS and many MS/DOS machines. No redistribution fees. $47 U.S. MN residents add 6% sales tax. BARTSUBS by BARTSOFT
3210 Lake Shore Boulevard
Wayzata, MN 55391
(612) 473-4032

REVISION CONTROL SYSTEM
The Software Revision Management System™ stores all versions of source/documentation in a single ASCII file. Allows retrieval of any version of source and application of changes, while recording when, why and where changes were made with no duplication of common code. MS/PC-DOS 2.0 $69.95 + .$5 s/h. (MN + 6%.) MC/Visa.
QUILT™ COMPUTING
7048 Stratford Rd.
Woodbury, Minnesota 55125
(612) 739-4650

BASIC TOOLS
Screen Development Aid $29
- Create screens quickly (40/80)
- TEST screens prior to coding
- Automate screen functions from BASIC SORTSUB—Sorts any array $19
- Sort a 1000 element array in 3 secs.

Interp. & Compiled BASIC, DOS 1.1-3.0 IBM PC, PCjr, XT, AT and compatibles
DEPENDABLE SYSTEMS
P.O. Box 191090
San Diego, CA 92119
(619) 287-3391

SoftRoms™
High Performance Graphics Tools. FAST; 200K pixels/sec avg. TIGHT; 14K. COMPLETE; 2/3D-Poly Lines, Bit Blocks, Windows, Fill, Arc, Icon, Character, String, Macros over 100 functions. Plus; assembler hooks, multiple language & hi-res boards, icon editor, memory draw & print, mouse & keyboard touch, type faces. Basic kit $70.00.
AFFINITY, INC.
126 Normal Ave.
Buffalo, New York 14213
(716) 882-5077

CROSSREF/BASIC & PASCAL
Debug programs, customise listings, control evasive variables! Our utilities will list your source, cross-reference variables in sorted order, and list procedure/function names. Available for Microsoft® Basic and Turbo Pascal® at $29.95 each or receive both for $49.95. All orders add $4.00 S&H. Check or Money Orders (Inquiries invited) to:
EXECUTIL
PO Box ### (12516)
Albany, N.Y. 12212

C TOOLS—MAKE
MAKE automatically keeps track of the dependencies between modules and include files. When you edit a file, you need only invoke MAKE to make sure that all the necessary modules are correctly compiled and linked and that object libraries are updated. Includes other useful tools. $40.00 check or COD.
INFOSOFT, Inc.
9020 Perryvale Rd.
Balto., MD 21236
(301) 256-2631

C PROGRAMMERS
Professional Software Development Tools
— Cross Module Cross Reference
— Single Module Cross Reference
— FindText—all occurrences of a given string in specified files (like GREP)
/*ALL SOURCE CODE INCLUDED (Pure K&R C)*/
MSDOS, CCPM, UNIX, MacIntosh, others
$165.00
SMITH & SMITH ASSOCIATES
P.O. Box 160
Hunt Valley, MD 21030
(301) 666-8129

**** PASCAL'S FRIEND ****
PASCAL'S FRIEND v2 contains source code for use with IBM PC Turbo Pascal: 1-2-3 style menu routines, keyboard handling, save & restore screens, read disk directory or any track & sector, system clock and calendar routines, write strings in any attribute, DOS function calls, MORE! $49.95 check, MO, Visa, MC
J.S. COMPUTING
815 N. 12th St. Suite 5
Allentown, PA 18102
(215) 821-9020

BTREES IN C WITH SOURCE
The Softfocus B-Tree Library is a complete set of data management functions written in K&R C. We support random and sequential access, multiple indices per data file and up to 16.7 million keys per index. Also, our software is customizable because we supply full source code. Manual & examples included.

B-Trees
$77.00

SOFTFOCUS
1277 Pallatine Drive
Oakville, ONT. CANADA L6H 1Z1
(416) 844-2610

SOURCE DEBUGGER FOR LATTICE C
The MSD DeBugger™ is the last, and perhaps final, word in programming assistance for Lattice C users. C Debugger produces a high-level view of C programs via function names line numbers, variable names and C data types, plus a low-level view of machine addresses and instructions for testing assembler language functions. More features include:
- All documentation is prepared for programmers.
- Online help screen throughout the process.
- Capability to single step through your program.
- Set break points, examine registers and variables.

$165.00 + $3.50 shipping VISA/MC

Cdebugger
a high level language debugger

MSD
214 1/2 West Main St.
St. Charles, IL 60174
(312) 377-5151

TECH BOOK

SOFTWARE GENERAL

C Users' Group

PUBLIC DOMAIN SOFTWARE IN C

Over 45 volumes of public domain software in CP/M & MS-DOS formats.

- editors & compilers
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details.
THE C USER'S GROUP
415 E. Euclid Box 97
McPherson, KS 67460
(316) 241-1065

EXCITING SOFTWARE GIFTS!

After a successful 1984 season GREETING-WARE® IS BACK with sing-along music plus software goodies. Ideal for Christmas and Birthdays. We have an exciting line of attractively priced software gifts for your favorite IBM PC user.
For our FREE catalog, write to
ROXBURY RESEARCH INC.
RD 1, Box 171A, Roxbury, NY 12474
or call NYS: 607-326-4070
Outside NYS, call toll free 800-DIALROX

SOFTWARE GRAPHICS

TEKTRONIX 4010 EMULATION

High resolution screen/printer graphics for the IBM PC. Full interactive capabilities with file transfer and cross-hair control. Utilizes PLOT 10, protocols. OFF-line review of graphics output. Supports IBM color/graphic or Hercules cards. Easy to use 4010 emulation at an affordable price. $80.00. Demo disk $5.
Technological Systems Group
5044 Haley Court
Lilburn, GA 30247
(404) 923-4980

GRAPHICS LIB FOR TECMAR

TEK-MAR lets you do high-res graphics on your TECMAR Graphics Master. TEK-MAR is a library for use with MS Fortran. Features windowing, viewporting, clipping, axis rotation. Similar to Tektronix graphics. Includes screen dump/restore, Epson screen print, support for Hewlett Packard, Western Graphtec plotter. Requires MS-FORT 3.20, 320K, GMDEV.SYS (5.2). Price: $195.
ADVANCED SYSTEMS CONSULTANTS
18653 Ventura Boulevard, Suite 351
Tarzana, California 91356
(818) 990-4942

MetaWINDOW™

Advanced graphics toolkit provides Xerox Star/Apple MacIntosh style graphics on your IBM PC. Supports most popular graphics cards. Allows you to create pop-up menus, windows & icons; use proportionally spaced fonts; rubberband & drag lines, text or bitmap images; supports mouse/cursor tracking. Tightly optimized for use with Turbo Pascal, IBM Pascal, C, Fortran & Compiled Basic.
METAGRAPHICS SOFTWARE CORP.
444 Castro Street, Suite 400
Mountain View, CA 94041
(408) 438-5352—Telex: 4998586 METAGRAF

PLOTTER SOFTWARE

For the IBM and APPLE computers to drive the Hewlett-Packard and compatible plotters. The DGI TYPE SHOP will almost transform your plotter into a type setting system. The DGI ORGANIZATION makes it easy to display and update your organization chart. To order or for additional information contact:
DECISION GRAPHICS, INC.
P.O. Box 2776 PC
Littleton, CO 80161
(303) 796-0341

SOFTWARE/INTERPRETERS

C-Terp™-C Interpreter

A COMPLETE DEVELOPMENT ENVIRONMENT. Full K&R language and libraries; Complete built in screen editor;
compiler compatible, supports object library compiled with C86 or Lattice C or assembler;
Multi-module support, includes global search; High-level symbolic debugging; Batchmode; Incredibly fast linking and semi-compilation; Runs on IBM-PC Dos 2.X, 3.X, 192K and up. Price: $300 (Demo plus manual $45) MC, VISA. See our display ad for more info.
Gimpel Software
3207 Hogarth Lane
Collegeville, Pa 19426
(215) 584-4261

SOFTWARE/LANGUAGES

IBM PC PASCAL/DOS 3.0/1

For the serious PASCAL and DOS user. EXAMPLES PLUS provide a rich set of examples and directly usable PASCAL function/proc code for PASCAL with DOS 2.0/1. EXAMPLES PLUS 2 provides additional info for PASCAL 2 with DOS 3.0. EXAMPLES PLUS 3 shows the differences and glitches in DOS 3.0/1, plus info on TopView, networking, extended graphic board and the AT hardware. Priced at $30 each, 2 for $48, 3 for $60. (CA add 6%).
JATO INC.
24861 Alicia Parkway, Suite 272-B
Laguna Hills, CA 92653

PASCAL-TO-C-TRANSLATOR

Convert UC5D Pascal, MT+Pascal, and others to K&R C. Handles nested procedures, intrinsic functions, separately compiled units and modules, all data types including long integers. Requires 256K IBM PC/XT/AT. Send for free samples or send us up to 500 lines of Pascal and we will convert it for you FREE. Licensed source and executable code $5,000, conversions 25 cents/line.

TGL INC.
4400 Sulphur Springs Road
Corvallis, OR 97330
(503) 745-7476

SOFTWARE/OPERATING SYSTEMS

pcSHARE MULTI-USER O/S

pcSHARE allows your IBM-XT or compatible to support up to 4 users running 123, dBASE, WordStar, etc on inexpensive serial CRTs. For the software developer, pcSHARE efficiently runs compiled Basic, Pascal & C programs with full DOS 3.0 record locking. No risk 30 day money back guarantee.
DIGITROL COMPUTERS, INC.
440 Phillip Street
Waterloo, Ontario, Canada N2L 5R9
(519) 884-4541

CX/PC C EXECUTIVE

Real-time, priority task scheduler. Functions to queue inter-task messages and pointers. Example tasks. Interrupt-driven timer, serial and parallel port interfaces. Runable demonstration programs. DeSmet C source included; no license for generated programs. Described in June '84 IEEE Micro article. $75. Brochure.
INTR-SOFT CO.
P.O. Box 351
Bedford, MA 01730
(617) 369-6242

QUADDRIVE

Extends PC-DOS to support 80 track 96tpi drives for a total of 806K per disk. Mix and match 48 and 96tpi drives and optionally boot from a 96tpi drive. Reads/writes/formats 48tpi in a 96tpi drive. Step speed (down to 2msec) is set for each drive. Requires DOS 3.x. $U49.00 Free Air Post. CHECK/MC/VISA. Dealer inquiries invited.
Digital Business Systems
P.O. Box 33
RIVETT ACT 2611 AUSTRALIA
011-61-62886520

SOFTWARE/SECURITY

SECURE AT/XT/PC

Control system access, data access! FiXT/S. Control system boot for most popular XT/PC hard disk controllers. Vfeature for AT-and-XT-compatible HD controllers segments hard disk by volumes, controls access with passwords, supports hard disk expansion. $70-$130 + $3 shpg. plus CA tax.

Golden Bow Systems
P.O. Box 3039
San Diego, CA 92103
(619) 298-9349

SOFTWARE/SERVICES

TAPE TO DISK CONVERSIONS

Convert any 9 track magnetic tape to or from over 200 formats including 3½", 5¼", 8" disk formats and word processors. Formats available include IBM-PC, Apple, Altos, TRS 80 8" CP/M, Displayer., IBM Sys/??. Macintosh, Wang, Lanier, OS/6 and 300 more. Disk to Disk conversions also avail. Call for more info.
PIVAR COMPUTING SERVICES, INC.
47 W. Dundee Rd.
Wheeling, IL 60090
(312) 459-6010

SOFTWARE/SPREADSHEET

FORMAT—1-2-3

- **Loads ANY ASCII data into Lotus™!**
- 1-2-3 and Symphony Utility!
- Send spreadsheets using E-Mail!
- On-Line Help System!
- Runs on PC, XT, AT, and lookalikes!
- Also TI Professional, AT&T
- Extended Warranty Available!
- DOS 2.0 + with 64K, 1 drive
- $150, MC, VISA, CK, COD

RELIABLE SOFTWARE COMPANY
P.O. Box D
Titusville, FL 32781
(305) 267-2043
(800) 235-6646 Ext 581 (nat)

TECH BOOK

Software/Stock Analysis

STOCK OPTION TOOL KIT
Use Monte Carlo analysis and random walk theory on stock option strategies. The kit has 50 pages of documentation, bibliography and 11 programs to evaluate covered calls, naked puts and calls, and various spreads. Price $199. Demo disk with one program $25. IBM PC, PC XT, 128K, DOS 2.0+ check or MO.
EXATECH SOFTWEAR
939 Pulpit Rock Circle North
Colorado Springs, CO 80907
(303) 599-0280

PC-STOCK GRAPHIC ANALYSIS
Graphic trend analysis aiding in technical stock, bond, and commodity analysis. Includes "ON BALANCE VOLUME" and moving average. Ideal for the small investor. PC-STOCK comes complete with source code and examples. Requires IBM-PC, 128k, DOS 1.1-2.1 and hi res graphics. $30 + $3 shipping.
SUNRISE ENTERPRISES
P.O. Box 2610
Goleta, CA 93118
(805) 968-4170

Software/Typesetting

HIGH-TECH TYPESETTING
Transmit your text via toll-free lines directly to our fully automated typesetting system. $2. per K characters with a $5. minimum. Same day service. 200 typefaces in sizes up to 72 point. Send $15. + $3. shipping for our 220 page guidebook, or call toll free and use your MC, VISA or AMEX.
INTERGRAPHICS INC.
106A South Columbus Street
Alexandria, VA 22314
(800) 368-3342 or (703) 683-9414 in DC area.

Software/Utilities

DISK MECHANIC
THE ULTIMATE Floppy Disk Backup & Repair Utility. Can back up ALL software protected disks written on the IBM PC. Works manually or automatically. Files or sectors can be restored, searched, examined & changed. Checks disk drive speed. req. IBM PC, DOS 1.1 (128K)/2 (192K) + 64K if only 1 floppy drive. $73 ppd. USA
MLI MICROSYSTEMS
PO BOX 825, Dept. TB
Framingham, MA 01701 USA
(617) 926-2055 for info MC/VISA

****DIGICON PRINT PACKAGE****
Want SIDEWAYS printing for SPREADSHEETS? How about BANNNER, PAMPHLET, LETTER, and BOOKLET printing too? Variety of fonts. Graphics characters. Character editor to MAKE YOUR OWN FONTS or SYMBOLS! Quick printer setup program too. Easy menus, online help. IBM/Epson printers, DOS, 128K. $49.95 (PA+6%).
DIGITAL CONCEPTS, INC
POB 8345 Dept. A
Pittsburgh, PA 15218
(412) 823-8314

NOBLINK™
ErGoNoMiCs! Eye strain? With just a keystroke you can change your blinking cursor into a friendly solid block reverse video cursor. If you wish you can change it back and forth (Blink/No blink) right in the middle of a program, such as your word processor. Give your eyes a rest. $29.95 plus $5 s&h.
NOSTRADAMUS
5320 South 900 East
Salt Lake City, UT 84121
(801) 261-0769

IBM USES CPM DISKS!!
Uniform lets you use CPM disks in your IBM B: drive while running any PCDOS/MSDOS program. Read and write to the CPM disk directly or transfer files to IBM format. Over 70 formats also 8" and 96 TPI. IBM PC, XT, AT. Call for details. Dealers invited. (Not for Apples). $59.95 + $5 S&H. Uniform by Micro Solutions.
BLUE HERON
RR 1, Box 128 E
Maple Park, IL 60151
(815) 895-5602

Hard Disk DIRECT ACCESS™
The ULTIMATE utility for any hard disk user. DIRECT ACCESS is a powerful hard disk management tool. Organizes your software programs into a "user defined" menu system. Features attractive screen display, rapid access into an application using a single keystroke & easy to learn. Also handles batch files & complicated DOS commands. Order today. Only $59.95 + $5.00 shipping & handling. MC/VISA.

DELTA TECHNOLOGY

DELTA TECHNOLOGY
P.O. Box 1104
Eau Claire, Wisconsin 54702
(715) 832-0958

AVOID FRUSTRATION!
PC-CATLG—easy to use, reliable, Diskette Management Utility. Directory info for EACH file is read/stored in a master catalog with a 40 char user comment. Functions: ADD/CHG/DEL/SRCH/UPD/DISPLAY/PRT/LABELS. Chg VOL labels. Req. DOS 1.1/2.X, 256K, 80 mono/color, 2 DSDD drvs (SSDD add $3), parallel printer opt. $25.
Oregon-microTech
1717 Magnolia Ave.
Medford, OR 97501
VISA/MC (503) 773-8327

SPEED UP YOUR DISPLAY!
FANSI-CONSOLE™: faster screen writing thru PC-DOS or BIOS (1.3-3.0+ times), larger X3.64 subset than ANSI.SYS, VT100/52 keys, more type-ahead, auto inactivity screen disable, eliminate scroll blink for some adaptors, window support, keyboard breakpts. Software & printed manual $50, or shareware disk $25.
HERSEY MICRO CONSULTING, INC.
P.O. Box 8276
Ann Arbor, Michigan 48107
(313) 994-3259

ALL DISK COMMAND SYSTEM
Impressive FILE HONEY™ system gives total access to subdir with 4 arrow keys. Sharpen PCs/XTs. Pampers today's users. Sorts, organizes, ID's, locates, scans-FAST-without memorizing subdir names. New tree access! Sophisticated wildcards & dates. Extra print features you like! Handsome folio $50—Demo $15. 128K/DOS 2.X.
THE JOE PHILLIPS COMPANY
9071 Metcalf, Suite 145-C
Shawnee Mission, KS 66212
(913) 341-3645

ASSEMBLY LANGUAGE PROS!
Sick and tired of those sloppy, misaligned listings of your masterpieces? ASMPRT will let *you* control where the tab stops are set for a slick, professional-looking program listing. Also allows you to compress the listing for those 80-column printers. Prints all other ASCII text files, too! Only $29.95.

Childs Systems Inc. USA

CHILDS SYSTEMS, INC.
P.O. Box 45212 Dept. 485
Baton Rouge, LA 70895
(504) 291-4167

WHAT IBM LEFT OUT . . .
The enhancer offers: fast flicker-free scroll • display review • pause at screen full • choose color for input • solid block cursor • oversized editable writable keyboard buffer displayed on screen • alarm clock with time display and chimes • menu entry of special characters. IBM PC & XT $50 check/MO Colo. res. add 3.6%.
GENERAL SOFTWARE
P.O. Box 3272 Dept 12
Boulder, CO 80307-3272

COMPUTER SYSTEM LOG
Monitor activities on your computer with SYSLOG. The System Log Utility is a command driven nonresident DOS extension that allows automatic task recording. With SYSLOG, all of your computer time is noted, saved, and categorized. Reports printed for tax purposes or for monitoring PC usage. Only $34.95 + Sh. VISA/MC accepted.

SYSLOG
SYSTEM LOG UTILITY

Apex Resource, Inc.
23 Christine Court
Stormville, NY 12582-9612
(914) 221-2611

UNIVERSAL CROSS REFERENCER
One Cross Referencer does it all for BASIC, C, Pascal, FORTRAN, Modula2, COBOL, DBASE-II, Macro Assembler, etc. User-defined page titles, page numbers, file date/time. Unlimited symbol length, adjustable output margins, compressed print. Multiple input files, Full directory and path support. DOS 2.x, IBM PC/PC-XT, 128k, $39.95.
DALSOFT Engineering
3565 High Vista
Dallas, Texas 75234

DOS PATH COMMAND FOR DATA
Now Lotus, c-Base, WordStar etc. can access data files no matter where they're located on your hard disk, floppies or RAM disk. D-PATH lets you organize your directories conveniently for you, not just your programs. Requires only 4K. Works on MS-DOS 2.0, 3.0 $25 + $25/h Visa, MC, company P.O., check.
PERSONAL BUSINESS SOLUTIONS INC.
P.O. BOX 757 DEPT. T
Frederick, MD 21701
301-865-3376

AT/XT/PC HARD DISK EXPANSION
Replace hard disk with a bigger one, or add a second drive! Vfeature supports high capacity drives on standard AT, XT, and compatible hard disk controllers. Includes multiple volumes, security features, selectable clusters, keyboard lock. $80-$120 + $3 shipping + CA tax.

GOLDEN BOW SYSTEMS
P.O. BOX 3039
SAN DIEGO, CA 92103
(619) 298-9349

COMPRESS/ENCRYPT FILES
TRUE/Miser compresses/expands ASCII files to reduce storage requirements. Specifically designed for the compression of long text and source code files. Reduces file size by up to 80%. Optional data encryption protects valuable data from unauthorized access. No limit on file sizes. $19.95
GRW Software
P.O. Box 440007
Houston, Texas 77244
(713) 556-9878

DISPLAY ANY MS-DOS FILE
List ASCII and binary files. TRUE/View can be invoked as a pop-up or directly from DOS. Switch between ASCII and hex modes. View multiple files. Search for string patterns. Mark file positions for easy retrieval. Make small changes with edit feature. User configurable. $19.95
GRW Software
P.O. Box 440007
Houston, Texas 77244
(713) 556-9878

TECH BOOK

TRUE PATH UTILITY

Never misplace files again. TRUE/Path extends the MS-DOS PATH utility to search for data files as well as programs. Unlike other path programs, TRUE/Path uses the same path given by the MS-DOS PATH command to locate data files. Works with all MS-DOS software. A perfect for inexperienced users. $19.95
GRW Software
P.O. Box 440007
Houston, Texas 77244
(713) 556-9878

dBASE II CROSSREFERENCE

Crossreference dBASE II data fields and variables with the DBXREF command. DBXREF has full wildcard support. Output can be directed to the screen or printer. DBXREF will crossreference one, multiple or all memory variables and data fields in a program. DOS or CP/M, $27 + $3 shpg MC/VISA.
Edward Foster & Company
1 West Illinois Street
St. Charles IL 60174
(312) 584-2989

TURBO PRINT FILTER

Issue commands to EPSON/IBM Printers from Turbo Pascal's Editor via the DOS I/O pipe. Let Turbo's Editor be your WORD PROCESSOR. So easy—2 key strokes invoke a command!!
CONDENSED EMPHASIZED ITALICS
PAGE NUMBERS LINES PER INCH MARGIN
DOUBLE-WIDE DOUBLE STRIKE and more
$29.95 visa/mc Includes the source code
AEF
P.O. Box 928
Katy, TX 77449
(713) 391-8570

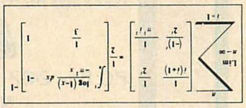
FKEY NOW AVAILABLE

The famous FKEY key processor is now running on PC, DOS/MS DOS. FKEY allows the redefinition of each key and key macros. No royalties on the run time package for product developers. Attractive dealer discounts and dealers are invited. Call or write for more information about FKEY. $39.95.
DITj
P.O. Box 60757
Sunnyvale, CA 94088
(408) 737-9000

SOFTWARE/WORD PROCESSING

FORMATH™ TEXT-FORMATTER

For technical texts. Sizes & places equations, matrices, ratios, integrals, big symbols, footnotes. Macros, multiple fonts, 158 Greek/math symbols. Automatic hyphenation, section/equation/reference numbering, indexes, table of contents. Dot-matrix, daisywheel, laser printers, plain/graphics monitors. $400. $50 for demo.



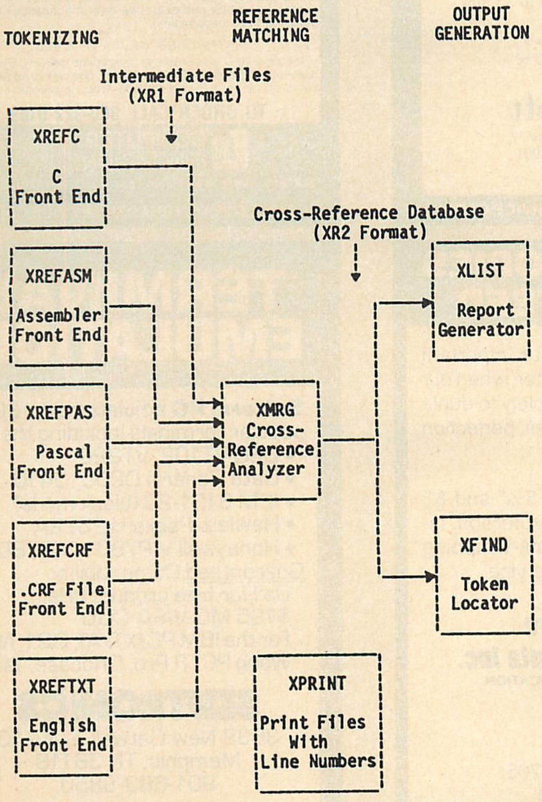
SHANTHA SOFTWARE, INC.
50 West 97th St. Room 11N
New York City 10025
212-222-7647
Touch-tone Toll-free: 950-1088-wait-FORMATH

VOLKSWRITER DELUXE

Features TextMerge™ for customized mailing lists; multi-lingual, engineering & scientific symbols; horizontal scrolling to 250 characters; document size 1 megabyte; on-screen page endings, underline, boldface, double-strike, strike-through, multiple fonts & colors—fewest keystrokes per function. Only $295.
LIFETREE SOFTWARE INC.
411 Pacific Street
Monterey, CA 93940-2790

PC BUYERS GUIDE

THE INDEPENDENT GUIDE TO PRODUCTS AND SERVICES FOR IBM Personal Computers. This master directory lists, describes, rates for ease of use, and tells you about the latest in software, hardware, accessories, supplies, books, services, et al. Everything for you and your IBM PC. It even includes an Index of Computer Camps for both adults and children. From the publishers of PC. An ideal gift at only $7.95 (+ $1.55 shipping/handling) to U.S. destinations; and $11.00 elsewhere. Residents of CA, CO, CT, DC, FL, IL, MA, MD, NJ, and NY add applicable sales tax. Checks or MasterCard or VISA acceptable (with number, expiration date and signature). U.S. funds only. Mail to:
PC BUYERS GUIDE
CN 1914
Morristown, NJ 07960



POLYXREF SYSTEM DIAGRAM

Now Generate Cross-Reference Listings Across Single or Multiple Source Files of One or More Languages

PolyXREF, *The Multi-Lingual Cross-Reference System* can cross-reference your source files even if you have used more than one source language. **PolyXREF** can do incremental cross-referencing — If you change only one module, you only need to run **PolyXREF** on that module and its information can be automatically merged with cross-reference information from other modules in your system.

Front-end Processors (Tokenizers) are available for Pascal, C, Assembler, and even English. Two Report Generators provide flexible listing generation.

PolyXREF:

Engine,.CRF Module, ALL four language front ends	219.00
*Engine,.CRF Module only	80.00
C Front End	49.00
Assembler Front End	49.00
Pascal Front End	49.00
English Front End	49.00

*To use **PolyXREF** you need to order the Engine and .CRF Module AND at least one language Front End.

To Order Call
1-800-547-4000
Ask for Dept. No. 310
Foreign & Oregon orders call (503) 648-3000
Send Checks, P.O.s To:
POLYTRON Corporation
P.O. Box 787 DS-310
Hillsboro, OR 97123
Add $5.00 Shipping To Total

VISA or MasterCard accepted

POLYTRON
High Quality Software Since 1982

CIRCLE NO. 159 ON READER SERVICE CARD

INDEX TO ADVERTISERS

IBM PC REPAIR

★ We have been repairing DEC equipment for over eight years and have expanded our product line to include the IBM PC. ★

System Board (PC/XT)	$110.00*
Monochrome Display Adapter	75.00
Color Graphics Adapter	75.00
5¼ Disk Drive Adapter	50.00
5¼ Disk Drive	50.00*
Printer Adapter	50.00
Keyboard	50.00*
Async. Adapter	50.00
SDLC Adapter	50.00*
32/64 K Expansion Option	50.00
Power Supply (PC)	60.00
Power Supply (XT)	75.00
Joystick Control	30.00

IBM AT (repair capability available soon)

* extra charges for special components may apply

All repairs include:

- 100% customer satisfaction
- 5 day turnaround
- **120** day warranty — *tops in the business!*
- Complete system testing

Electronic Service Specialists Ltd.
N92 W14612 Anthony Drive • Menomonee Falls, WI 53051
Phone (414) 255-4634

Your DEC and IBM Repair Specialists!

CIRCLE NO. 120 ON READER SERVICE CARD

OVERCOME
FORTRAN | PASCAL | C
LIMITATIONS WITH

$129
NO LICENSE FEE

NO LIMIT

Ver. 2.0 For:
MS FORTRAN/Pascal/C
IBM Professional
and R-M FORTRAN
SuperSoft FORTRAN

A library of over 90 Assembler routines transform FORTRAN, Pascal and C language compilers into the flexible, responsive, complete languages needed for the microcomputer environment. Hundreds of NO LIMIT owners are creating highly interactive software systems, often utilizing existing mainframe code and saving time and money in the process. Ver. 2.0 features include:

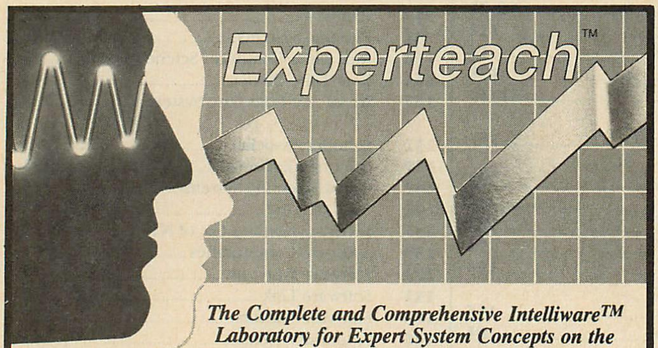
EXTENSIVE GRAPHICS	(Get, Put, Paint, Color, Dot, Line, Box, Circle, Ellipse, Large Characters)
FULL SCREEN CONTROL	(Windows, Cursor, Read/Write Screen)
STRING MANIPULATION	(Match, Compare, Concatenate/Extract, Pack, Justify, Zero Fill)
KEYBOARD CONTROL	(Read Key During Execution, String Read With Edit)
FILE MANAGEMENT	(Exist?, Rename, Delete)
COMMUNICATIONS	(Full Interrupt Driven to 9600 Baud, Set Com Line, Send/Receive, Line/Modem Status)
INTERRUPTS	DOS Execution, Program Execution
OTHER FEATURES	(Command Line Read, DOS 3.0 Directories, Peek, Poke, Random Numbers, System Status, etc.)

And to complement NO LIMIT, the I/O PRO development system allows creation and editing of FORTRAN/Pascal/C callable screens. This word processor type system pays for itself several times over on your first project. $260.

For immediate solutions to your programming needs call
(800) 562-9700
(512) 251-5543 (Texas)
M | E | F Environmental Inc.
P.O. Box 26537 Austin, Texas 78755

©Trademark of Microsoft Inc., Supersoft Inc., IBM Corporation, Ryan-McFarland and Microsoft Inc., respectively.

CIRCLE NO. 174 ON READER SERVICE CARD



The Complete and Comprehensive Intelliware™ Laboratory for Expert System Concepts on the IBM Personal Computer.

Experteach Includes:

- Comprehensive Introduction to Expert System Concepts.
- On-line Tutorial Describing the Operation of Expert Systems.
- Lisp Based Expert System Tools with Source Code.
- Prolog Based Expert System Tools with Source Code.
- dBASE II™ Based Expert System Tools with Source Code.
- Pascal Based Expert System Tools with Source Code.
- Complete Lisp Interpreter for the IBM PC®.
- Complete Prolog Interpreter for the IBM PC.
- Comprehensive Case Studies of Several Major Expert Systems.
- Comprehensive bibliography on Expert Systems.

Experteach is a comprehensive guide to Expert System technology consisting of a uniquely integrated collection of Expert System tutorials, case studies, on-line teaching programs, Expert System building tools with source code and Artificial Intelligence languages. Experteach is based on extensive experience in teaching Expert System concepts in association with IEEE, ACM and the Continuing Education Institute.

Experteach introduces you to Expert System technology by allowing you to build Expert Systems and to experiment with a variety of Artificial Intelligence tools and languages on the IBM PC.

Experteach includes eight rule-based Expert System shells with source code implemented in Lisp, Prolog, dBASE II and Pascal. Each language has been used to implement both a forward chaining and a backward chaining Expert System shell with a built-in rule editor, inexact inference and how & why explanation facilities.

Experteach includes a comprehensive Lisp interpreter and a complete Prolog interpreter with DEC-10 Prolog syntax. Experteach requires only 256K of memory.

Intelliware, Inc., 4676 Admiralty Way Suite 401 Marina del Rey, CA 90291 (213) 827-1334

☐ Introduction to Expert System Concepts, On-line Tutorial, Case Studies, Bibliography and Pascal Based Tools $99.00.

☐ Lisp, Prolog or dBASE II Based Tools $99.00 each.

☐ Complete Experteach System with Lisp and Prolog Interpreters. $475.00 Check, Money order, Visa or Mastercard. $9.00 for postage and handling. California, 6.5% tax.

Intelliware, Inc.

Experteach and Intelliware are trademarks of Intelliware, Inc. IBM PC is a registered trademark of IBM. dBASE II IS A TRADEMARK OF Ashton-Tate.

CIRCLE NO. 126 ON READER SERVICE CARD

SCIENTIFIC POP UP CALCULATOR

How Many Times Have You Had To Reach For A Hand Held Calculator While using Your PC?

WELL REACH NO MORE!

Introducing The KSH-1 Full Function Scientific Pop Up Calculator - The Calculator With A Difference.

No longer do you have to contend with just 4 functions on a pop up calculator. The KSH-1 calculator provides all the functions available on the HP-11C plus more:

- Calculations to 18 digits (17 displayable)
- Ability to insert result directly in foreground program
- Graphically presented (no references cards needed)
- Program steps and error messages displayed in English
- All stack registers viewable simultaneously
- Number base conversions (hex, binary, octal, decimal)
- Save/recall programs to disk
- All KSH-1 calculator software is RAM resident (-35K)

Requirements: IBM PC, XT, AT, or true compatible with 64K (128K recommended); monochrome or color, 1 drive, DOS 2.0 or later

Price: $49.95 (either plain or 8087 version)
$59.95 (both versions)

Terms: $3 shipping, Visa, Mastercard, M.O., check
Tenn. add 7.25% sales tax

All orders include free Microsoft Mouse support version.
(MOUSE NOT INCLUDED)

KSH

K Software House
Rt. 2, Box 83B1
Unionville, TN 37180
(615) 294-5090

CIRCLE NO. 146 ON READER SERVICE CARD

INDEX TO PRODUCTS

1 MILLION BYTE DISK STORAGE FOR $50

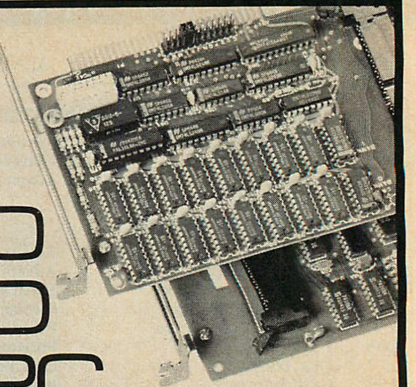
- MCFORMAT software adds 15% to 50% to hard disk capacity.
- Gives full control over disk format options including standard DOS configuration
- Requires no special training or repeated operations
- Occupies a small amount of RAM alongside DOS
- Fully DOS compatible
- Requires IBM PC (or 100% compatible) with monitor, 128K RAM, and DOS 3.0 or 3.1

To **Microcomputer Concepts, Inc.**
phone (206) 236-2300 (800) 722-8088
9715 SE 43rd Street, Mercer Island, Washington 98040

☐ YES! I want my hard disk to hold a million bytes more
Please send me ______ copies of MCFORMAT at $50.00 each
☐ My check drawn on a U.S. bank is enclosed
☐ VISA ☐ MASTERCARD ________________ Expires ______
I understand I may request a full refund within 30 days if not satisfied
Please ship to
Name ________________
Street ________________
City ________________ State ________ ZIP ______
Phone ________________ Signed ________________
(Washington state residents please add $4.05 sales tax)

CIRCLE NO. 184 ON READER SERVICE CARD

TC-50 AND TC-PC



9-TRACK TAPE CONTROLLERS AND ½" TAPE SUBSYSTEMS

- Operates with IBM-PC, -XT and -AT
- Complete interchangeability with other 9-track systems
- Installable device drivers for IBM XENIX and MS-DOS
- Complete data interchange utilities provided
- Data transfer rates up to 904,000 bytes/second; data densities up to 6250 bytes/inch
- Prices from $880 for controllers; from $2995 for complete tape systems

OVERLAND DATA, INC.
5644 Kearny Mesa Road #A
San Diego, CA 92111
Tel. (619) 571-5555
Telex 754923 OVERLAND

CIRCLE NO. 185 ON READER SERVICE CARD

Enhance your IBM™ PC Network

1. PcPhone™ — the electronic 'chat' system for interactive communication over the IBM PC Network
   - provides full duplex communication with split screens
   - can 'POP-UP' on user defined activation key(s)
   - is easy to learn and remember, easy and quick to use
   - offers directory assistance to aid in placing a call
   - is installed as a RAM resident program
   - provides incoming call notification by ringing bell even while current application is running
   - runs with or without the IBM PC Network Program
   - runs under Topview or under DOS 2.x / 3.x
   - is a small implementation with realtime performance
   - contains a convenient on-line 'help' facility
   - has a 'login' feature allowing you to move from one PC to another and still receive your calls

Minimum PC requirements: IBM PC/XT/AT or compatible, 64KB of extra memory, floppy disk, and IBM PC Network hardware

PcPhone price: 3 PCs = $49, 8 PCs = $199, 16 PCs = $299, 32 PCs = $399, over 32 PCs = call; $10 for manual only

For quickest response, order with Visa/MC by calling
800-621-2283 ext 912
In TX: 800-442-5321 ext 912

Or send check/money order to address below. TX orders add 6 1/8% sales tax.

2. Services
   - Workshops for network application programmers
   - Custom programming to convert single user software into multi-user software for use on the IBM PC Network
   - Establish network administrative procedures
   - NETBIOS development for your network hardware
   - Network design and analysis using performance model

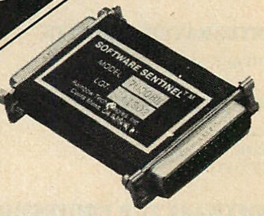
PC Network Specialists, Inc.
Dept. T59, 13109 Chandler Drive
Dallas, TX 75243
(214) 373-2386

IBM is a trademark of International Business Machines Corporation.
PcPhone and CommShare are trademarks of PC Network Specialists, Inc.

CIRCLE NO. 156 ON READER SERVICE CARD

SOFTWARE SENTINEL™

MUCH MORE THAN JUST PROTECTION



Because its original objective was to protect software from unauthorized use, it's called the SOFTWARE SENTINEL. But maybe we'll change the name to *REVENUE GENERATOR*, since we've discovered its true value to our customers. But whatever we call it, everyone ends up with numerous benefits from this hardware key with unlimited "locks":

SOFTWARE DEVELOPER (PC, XT, AT)

- Prohibits unauthorized use of software
- No need for copy protection
- Easily controlled site licensing
- Unbreakable protection
- Lower cost of distributing updates
- Longer product/revenue life cycle
- Trial usage/rental of software programs

USER (PC, XT, AT)

- Unlimited backup copies
- No floppy required with hard disk
- Transportable
- Pocketsize
- Transparent
- Data security

EVALUATION KIT AVAILABLE ($250 REFUNDABLE)

RAINBOW TECHNOLOGIES, INC.
17971 SKYPARK CIRCLE SUITE E, IRVINE, CA 92714 (714)261-0228

CIRCLE NO. 157 ON READER SERVICE CARD

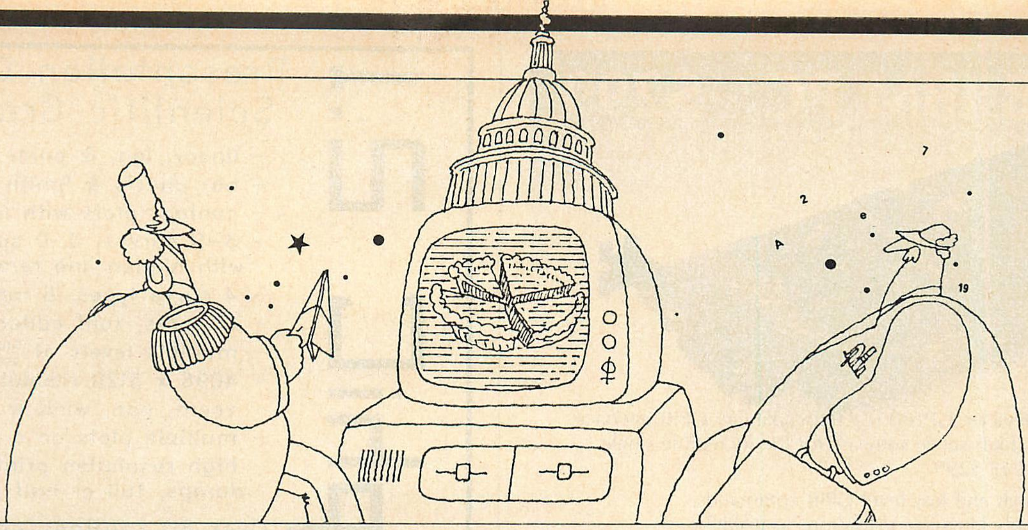


ILLUSTRATION • DAVID POVILAITIS

SEPTEMBER

September 1
Call for papers: 10th World Computer Conference
Dublin, Ireland
(September 1-5, 1986)
Sponsor: International Federation for Information Processing
Contact: Philip H. Dorn, Dorn Computer Consultants, Inc., 25 E. 86th Street, New York, NY 10028; 212/427-7460

September 3-6
OASI Third Annual Conference: The Integrated Office—How Soon?
Bloomington, MN
Sponsor: Office Automation Society International
Contact: OASI, 15269 Mimosa Trail, Suite B, Dumfries, VA 22026; 703/690-3880

September 4-6
IFIP Working Group 8.1 Working Conference on Environments to Support Information System Design Methodologies
Bretton Woods, NH
Contact: Peter C. Lockemann, Institut fur Informatik, Universitat Karlsruhe, Zirkel 2, 7500 Karlsruhe 1, West Germany; 49/0721/608-3968

September 5-7
3rd Personal Computer Faire
San Francisco, CA
Contact: David Small, Computer Faire, Inc., 181 Wells Avenue, Newton, MA 02159; 617/965-8350

September 9-11
Federal Computer Conference
Washington, DC
Contact: The Federal Computer Conference, P. O. Box N, Wayland, MA 01778; 617/358-5356 or 804/747-6448

September 10-12
Compint 85: First International Conference on Computer-aided Technologies
Montreal, Canada
Sponsor: IEEE-CS, ACM SIGDA, and SIGART
Contact: Stephen G. Leahey, P.O. Box 577, Desjardins Postal Station, Montreal, PQ H5B 1B7, Canada; 514/870-8842

September 10-13
9th Data Communications Symposium
Whistler Mountain, British Columbia, Canada
Sponsor: ACM SIGCOMM, IEEE-CS, IEEE Communications Society
Contact: W. P. Lidinsky, Room 6B 309, AT&T Bell Laboratories, Naperville Wheaton Road, Naperville, IL 60566; 312/979-6817

September 11-13
CAD 2001: The Countdown
London, England
Contact: CAD Seminars, Inc., 150 E. Riverside, Suite 400, Austin, TX 78704; 512/445-7342

September 16-18
C Language Seminar
Cambridge, MA
Contact: Beatrice Blatteis, CL Publications, 131 Townsend Street, San Francisco, CA 94107; 415/957-9353

September 17-18
Workshop on Simulation and Test Generation Environments
San Francisco, CA
Contact: IEEE, P.O. Box 639, Silver Spring, MD 20901; 301/589-8142

September 17-19
Sixth Annual SOFTWARE/expo
Dallas, TX
Contact: SOFTWARE/expo, Suite 205, 2400 E. Devon Avenue, Des Plaines, IL 60018; 312/299-3131

September 18-20
UNIX EXPO
New York, NY
Contact: Don Berey, National Expositions Co. 14 W. 40th Street, New York, NY 10018; 212/391-9111

September 21-22
7th Annual FORTH National Convention
Palo Alto, CA
Contact: FORTH Interest Group, P. O. Box 8231, San Jose, CA 95155; 408/277-0668

September 24-26
AI/Europa: Artificial Intelligence and Fifth Generation Computer Technology Conference and Exhibition
Wiesbaden, West Germany
Contact: Jim Hay, Tower Conference Management Co., 331 W. Wesley Street, Wheaton, IL 60187; 312/668-8100

September 26-29
8th Northeast Computer Faire
Boston, MA
Contact: David Small, Computer Faire, Inc., 181 Wells Avenue, Newton, MA 02159; 617/965-8350

OCTOBER

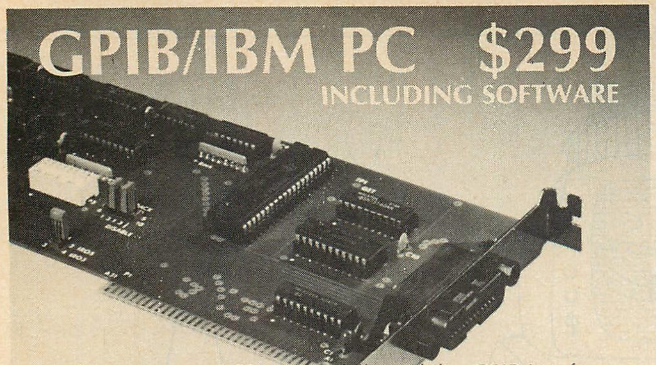
October 2-4
First International Conference on Optical Storage Technology
Tokyo, Japan
Sponsor: TOC and USACO Corporation
Contact: Technology Opportunity Conference, P.O. Box 14817, San Francisco, CA 94114; 415/626-1133

October 3-5
UNIX Systems Expo 85
Boston, MA
Contact: David Small, Computer Faire, Inc., 181 Wells Avenue, Newton, MA 02159; 617/965-8350

October 5-8
5th Symposium on Small Computers in the Arts
Philadelphia, PA
Contact: Symposium, P. O. Box 1954, Philadelphia, PA 19105

October 7-11
Compsac 85
Chicago, IL
Contact: Compsac 85, P.O. Box 639, Silver Spring, MD 20901; 301/589-8142

GPIB/IBM PC $299
INCLUDING SOFTWARE



Real Time Devices' GP100 is a full capability GPIB interface that includes high-speed software for the incredible single quantity price of $299.

- Over 40 high and low level GPIB commands
- Fast assembly language extensions to BASIC
- DMA and interrupt capability
- Comprehensive tutorial/instruction manual
- No hidden software, cabling, or documentation charges
- Supports up to 15 devices
- FORTRAN, PASCAL, and FORTH extensions available

Even if you're not familiar with GPIB interfacing, the GP100 will have you controlling this powerful bus in no time at all, allowing you to focus on your application requirements.

So if you've been intimidated by the complexity or cost of other IBM PC/GPIB implementations call on us to help — you'll be pleasantly surprised.

Real Time Devices, Inc.

rtd

1930 PARK FOREST AVENUE
P.O. BOX 906
STATE COLLEGE, PA 16804
(814) 234-8087

CIRCLE NO. 165 ON READER SERVICE CARD

GraphiC 2.1

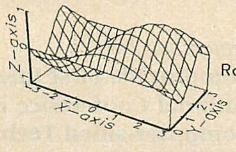
Presentation Quality Scientific Graphics

- linear, log, & polar plots
- bar charts & Smith charts
- contour plots with labels
- 3–D curves, 3–D surfaces with hidden line removal
- 4 curve types, 8 markers
- 12 fonts, font editor
- multiple levels of super/sub scripts
- 4096 x 3120 resolution
- zoom, pan, window plots
- multiple plots on a page
- high resolution printer dumps, full or half page

Over 80 routines can be called by your C program, SOURCE INCLUDED for private use only. $250. Demo $8.

256k, IBM & Corona PCs, DOS 2.xx
Epson, Okidata, Prowriter, Toshiba printers
Hewlett Packard, Houston, Sweet Pea plotters
Amdek, Tecmar, Hercules, IBM graphics boards
DeSmet, Lattice, C–86 C compilers

Ask about VTEK, our VT100/Tektronix emulator.



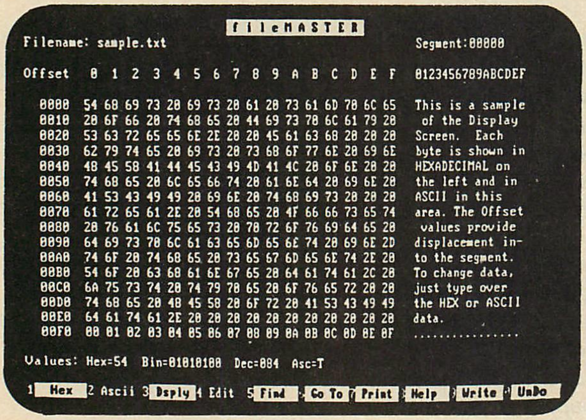
GraphiC 3–D Plot

Scientific Endeavors
Route 4, Box 79, Kingston TN 37763
(615) 376–4146

THIS AD WAS PRODUCED BY GraphiC™

CIRCLE NO. 187 ON READER SERVICE CARD

fileMASTER THE DISK UTILITY



FINALLY!! A Disk Utility for the IBM-PC that's fast and easy to use.

- FILE MODE
- DISPLAY
- PRINT
- MODIFY
- SECTOR MODE
- BROWSE
- SEARCH
- DOS 1.1 to 3.0

ONLY **$39.95** + $2.00 SHIPPING, CA + 6.5% TAX

J. L. SCHULLER & ASSOCIATES (818) 366-6934
14800 Rinaldi St., Suite 6, Mission Hills, CA 91345

CIRCLE NO. 232 ON READER SERVICE CARD

TURBO EDITASM

Introducing the first co-resident editor assembler for the IBM PC family. ***TURBO EDITASM (TASM)*** is significantly faster and easier to use than the IBM Macro-Assembler (MASM). Whether you are new to assembly language and want to quickly write a small assembly language routine, or are an experienced MASM user tired of waiting months to assemble large files, ***TURBO EDITASM*** will bring the excitement back to assembly language.

***TURBO EDITASM* IS MUCH FASTER:**

- How fast is ***TASM?*** The graph below shows relative assembly times for a 48K source file. For large files like this we blow MASM's doors off at 3 times their speed. For smaller 8K files we positively vaporize them at 6 times their speed.

  TASM (110 sec.)
  MASM (340 sec.)

- ***TURBO EDITASM*** is faster for the following reasons: **(1)** Written entirely in assembly language (unlike MASM). **(2)** Editor, assembler and source file always in memory so you can go instantly from editing to assembling and back. **(3)** Eliminates the time needed to LINK programs. Executable COM files can be created directly. (Also creates OBJ files compatible with the IBM linker).

***TURBO EDITASM* IS EASIER TO USE:**

TASM includes many other features to make your programming simpler.

- Listings are sent directly to screen or printer. Assemblies can be single stepped and examined without having to leave the editor.
- Access the built-in cross reference utility from the editor.
- Full support of 186 and 286 (real mode) instructions.
- Both Microsoft and 8087 floating point formats are supported. 8087 and 287 instructions supported directly without macros for faster assembly.
- Calculator mode: Do math in any radix even using symbols from the symbol table.
- Direct to memory assembly feature lets you test execute your code from editor.
- Coming soon: A coordinated symbolic debugger.

COMPATIBILITY: ***TASM*** is source code compatible with MASM and supports macros, records and structures.

Introductory Price $49
With .OBJ Capability $99

Include $5.00 shipping and handling. California residents add 6% Sales Tax.

Dealer inquires welcome

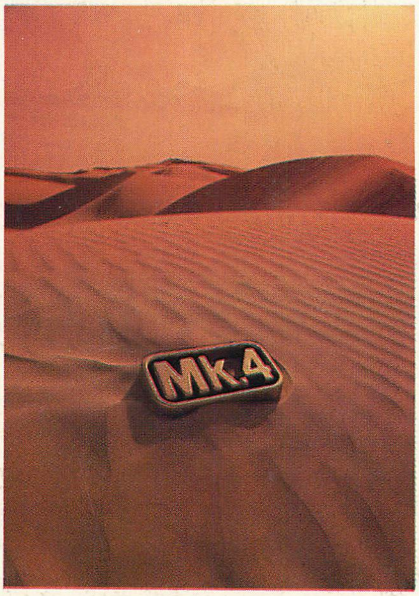
916-988-7426
118 Buck Circle, Box T
Folsom, CA 95630

Speedware™

IBM, Microsoft trademarks of IBM Corp., Microsoft Corp.

CIRCLE NO. 190 ON READER SERVICE CARD

New Generation Communications



Our new Crosstalk Mark 4 behaves just as reliably as the Crosstalk you've always trusted. But when you ask it for a bit extra, you're in for some surprises.

Up To 15 Concurrent Sessions

Mark 4 supports the X.PC multiple-session protocol, so it's capable of up to 15 concurrent communications sessions, each with the end-to-end error-checking needed for tomorrow's higher speed modems.

With more than one session going on at once, you need some way to keep track of them all. Crosstalk Mark 4 has that, too.

What You Get Is What You See

Mark 4 identifies each session with a "page" number. You can flick from one session to another with one keystroke. See each one full screen.

But if you'd like to keep an eye on more than one session at once, you can create windows — as many as you need in any size or shape — to display them all.

Menu? Or Command?

No matter how expert you are, Mark 4 is just your speed. It operates on command, or with a menu, or any combination of the two.

If you need help at any point in your command sequence, Mark 4 gives you suggestions that apply precisely to the task at hand.

Why Repeat Yourself?

If you make the same calls often, as most people do, Mark 4 can save you a lot of dull repetition. It has built-in command programs to call up and log in to most of the major information utilities.

But Mark 4 goes one step further. It can "memorize" any command sequence you perform, then repeat it that way any time you ask it to. You can't make programming much easier than that.

And Now, By Popular Request ...

— Mark 4 has a text-editor built in. You can create and edit files without having to leave Crosstalk.
— Mark 4 emulates the most popular terminals, including IBM 3101, DEC VT-52, VT-100, and the TeleVideo 900 series. Most other programs emulate one or two.
— In addition to X.PC, Crosstalk Mark 4 supports Kermit, Xmodem, and of course our own Crosstalk protocol.

How New Is New Generation Communication?

New enough for the advanced breed of modem that's already coming around the corner. New enough to give you the best high-speed, error-checked communication possible on noisy phone lines — or secure dedicated lines.

Finally, because Crosstalk is already the industry standard for small business computers, Mark 4 is at home in a broader universe than any other communications software.

Our Choice Is Your Choice

Naturally, we still offer our best-selling Crosstalk XVI program, for almost all business microcomputers. It has all the communications features that many users will ever need.

If you need more, Mark 4 obviously has more.

We'll even upgrade your copy of Crosstalk XVI to Crosstalk Mark 4, for a moderate charge. It's your choice.

Mark 4 may be new and improved and revolutionary — but it's still Crosstalk.

CROSSTALK® MK.4

MICROSTUF®
1000 Holcomb Woods Parkway / Roswell, Georgia 30076